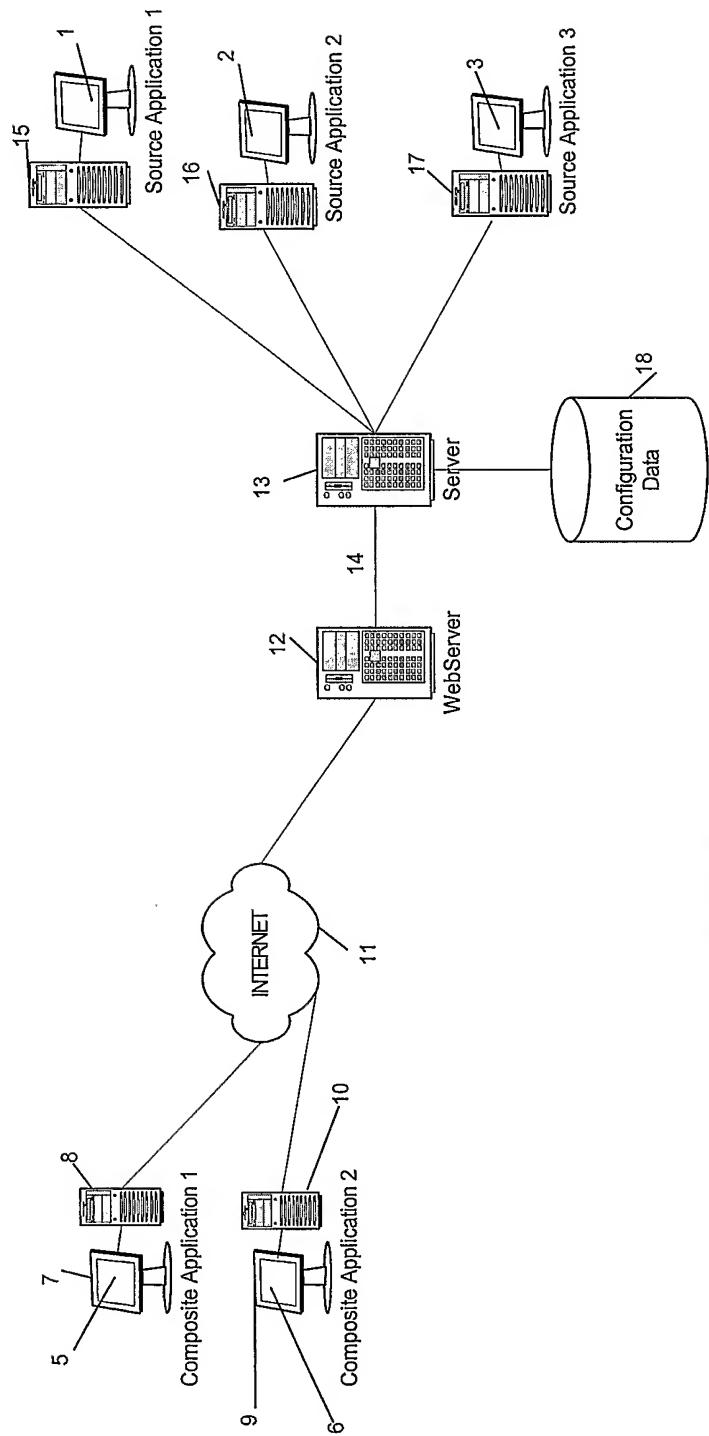


FIG 1

2/48

**FIG 2**

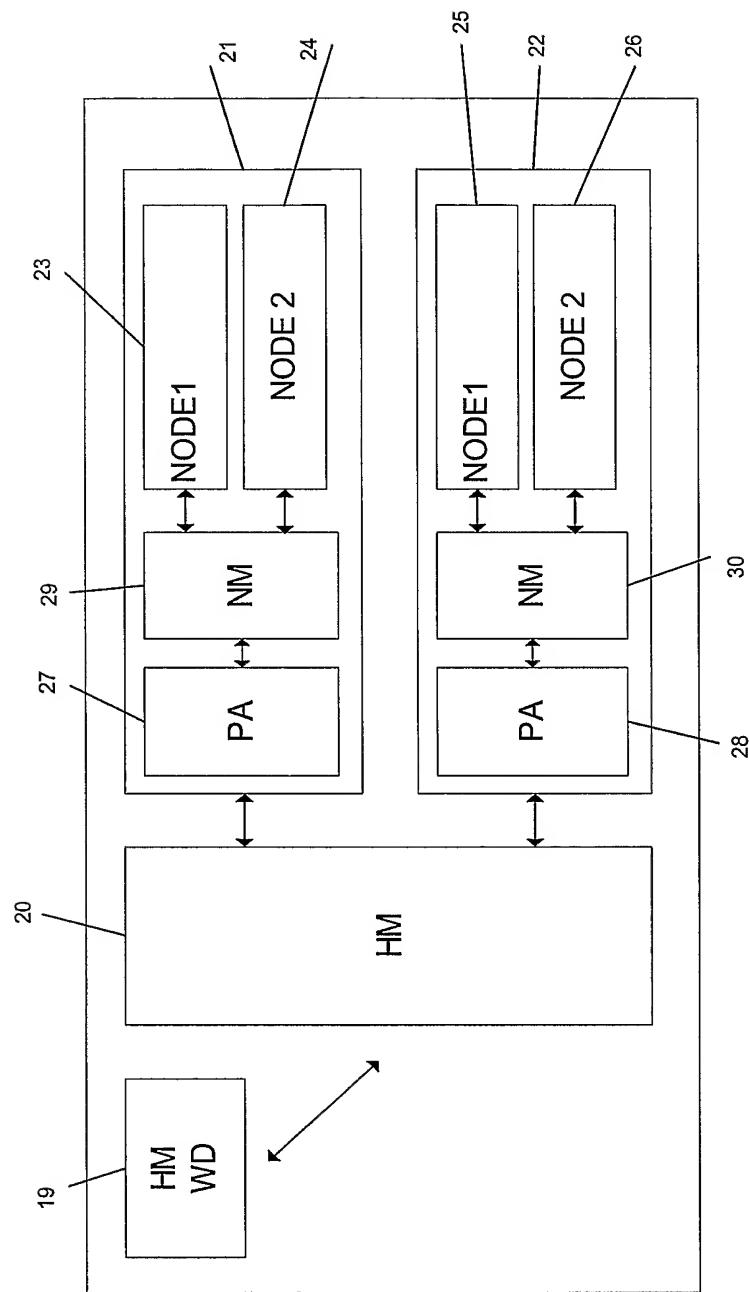
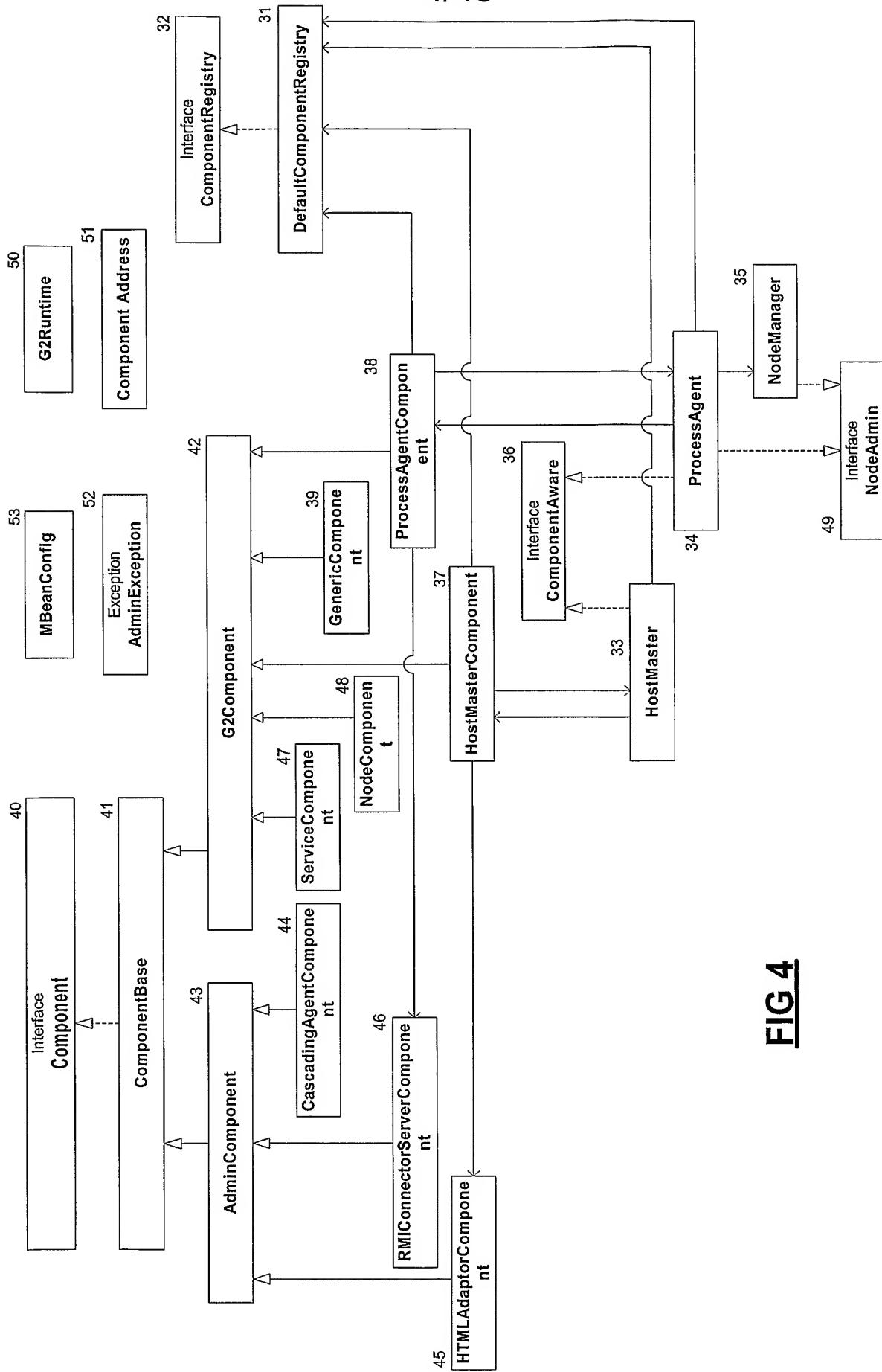


FIG 3

**FIG 4**

5/48

31

DefaultComponentRegistry	
<pre> -DefaultComponentRegistry minstance=new DefaultComponentRegistry() -MbeanServer mMyMBEanServer -String mG2MbeanDomain -String mDefaultDomain -String mHostIpString -String mProcessIdString -Hashtable mRegisteredComponent -DefaultComponentReistry() -void init() -void registerMBeanObject(String domain, Object obj, Hashtable nvp) -void unregisterMBeanObject(ObjectName target) -void CleanUpMBeanServer() -void registerComponent(Component c) -void unregisterComponent(Component c) -Component registerComponentAware(ComponentAware ca) -Component registerGenericComponent(Object o) -Component registerRmiConnectorServer(RmiConnectorServer o) -Component registerCascadingAgent(CascadingAgent o) -Component registrHtmlAdapterServer(HtmlAdapterServer o) -void cleanup() -Set findObjectNames(AddressMask am) +Hashtable getComponentProperties(Component c) +ComponentRegistry getInstance() +ComponentAddress lookupComponentAddress(Object o) +void waitForComponentRegistration(String objectname, int maxwaitseconds) +Component registerObject(Object o) +void unregisterObject(Object o) +void addNotificstionListener(ObjectName name, NotificstionListener listener, java.lang.Object.handback) +void invokeOnObject(AddressMask m, String method, Object[] params, String[] signature) +void invokeOnObject(Object o, String method, Object[] params, String[] signature) +void invokeOnObject(ComponentAddress addr, String method, Object[] params, String[] signaure) +Set findLocalObjectComponents(AddressMask props) +Set findGlobalObjectComponents(AddressMask props) </pre>	

FIG 4A

32

ComponentRegistry	
<pre> +Component registerObject(Object c) +void unregisterObject(Object c) +void invokeOnObject(AddressMask addr, String method, Object[] params, String[] signature) +void invokeOnObject(ComponentAddress addr, String method, Object[] params, String[] signature) +void invokeOnObject(Object o, String method, Object[] params, String[] signature) +void WaitForComponentRegistration(String objectname, int maxwaitseconds) +ComponentAddress lookupComponentAddress(Object o) +Set findLocalObjectComponents(AddressMask props) +set findGlobalObjectComponents(AddressMask props) </pre>	

FIG 4B

6/48

33

HostMaster	
-boolean misDebug	
-int mNextProcessId	
-Vector<ActiveProcessIds>	
-boolean mIsAlive	
-ComponentRegistry mCRegistry	
-HostMaster mTheMaster	
-HostMasterComponent mHotMasterComponent	
-String mProcessPropsString	
.	
+HostMaster()	
+int addProcess()	
+void destroyProcess(Integer pid)	
+void main(String[] args)	
+void shutdown()	
+void kill()	
+void SetIsDebug(boolean d)	
+boolean isDebug()	
+boolean isAlive()	
-boolean getBooleanArg(String token, String[] args)	
+void processLost(int pid)	
+Class getComponentClass()	
-void recover()	
-ShutdownMonitor	

FIG 4C

34

ProcessAgent	
-int mMyPid	
-ProcessAgentComponent mAgent	
-ProcessAgent mAgent	
-NodeManager mNodeManager	
-boolean mIsAlive	
-ComponentRegistry mCRegistry	
+ProcessAgent(int pid)	
+boolean isAlive()	
+void main(String[] args)	
+void addNode(NodeType nt)	
+void removeNode(NodeType nt)	
+void startNode(NodeType nt)	
+void stopNode(NodeType nt)	
+void destroyProcess()	
+void addNode(String nt)	
+void removeNode(String nt)	
+void startNode(String nt)	
+void stopNode(String nt)	
+int getProcessId()	
+Class getComponentClass()	

FIG 4D

7/48

35

NodeManager
-Vector mynodes
+void NodeManager()
+Node createNode(NodeType type)
-void destroyNode(Node n)
-void startNode(Node n)
-void stopNode(Node n)
+void destroyNode(NodeType nt)
+void startNode(NodeType nt)
+void stopNode(NodeType nt)
+Node getNode(NodeType type)
+Integer activeNodeCount()
+void recoverNodes()

FIG 4E

36

ComponentAware
+Class getComponentClass()

FIG 4F

37

HostMasterComponent
-String MBEANTYPE="HostMaster"
-String MYBEAQNDESCRIPTION
-int HEARTBEATRETRIES=1
-int HEARTBEATPERIOD=5000
-HostMaster mHostMaster
-HtmlAdapterComponent mHtmlAdapterComponent
-Hashtable mRemAgents
-ComponentRegistry mCRegistry
+HostMasterComponent(Object hm)
-CascadingAgent createCascadingAgent(String cascadingDomain, String hostname, int targetProcess
+String getType()
+void processAdded(int pid)
+void processRemoved(int pid)
+void preRegister()
+void preUnregister()
+void handleNotification(Notification n, Object o)
+void processRecovered(int pid)
#String getMBeaqnDescription()
#ModelMBeanOperationInfo[] getMBenOperations()

FIG 4G

8/48

38

ProcessAgentComponent
<pre>+String MBEANTYPE="ProcessAgent" -String MYBEANDESCRIPTION="" -ProcessAgent mProcessAgent -RMICConnectorServerComponent mRMIServer -ComponentRegistry mCRegistry</pre>
<pre>+ProcessAgentRegistry(Object pa) + void preRegister() +void preUnregister() +String getType() #String getMBeanDescription() #ModelMBeanOperationInfo[] getMBeanOperations()</pre>

FIG 4H

39

GenericComponent
<pre>-String MYBEANDESCRIPTION="Object" +GenericComponent(Object o) +String getType() #String getMBeanDescription()</pre>

FIG 4I

40

Component
<pre>+String IDPROP="id" +String TYPEPROP="type" +String HOSTPROP="host" +String PROCESSPROP="process" +String NODEPROP="node" +String SERVICEPROP="service"</pre>
<pre>+String getId() +String getType() +String getService() +String getNode() +String getAdminDomain() +Object getObject() +Object getMBean() +String getHostId() +String getProcessId() +void preRegister() +void preUnregister()</pre>

FIG 4J

9/48

41

ComponentBase
<pre> -Object mTheObject -String mObjectDescription -Object mTheMBean -ArrayList mMBeanOperationInfo -ArrayList mMBeanAttributeInfo -ArrayList mMBeanNotificationInfo -String mHostIp -String mProcessId +ComponentBase(Object o) +boolean isMBean(Object o) +String getId() +String getType() +String getService() +String getNode() +Object getObject() +void preRegister() +void preUnregister() +Object getMBean() #void setObjectDescription(String desc) #ModelMBeanOperatingInfo[] getMBeanOperations() #ModelMBeanAttributeInfo[] getMBeanAttributes() #ModelMBeanNotificationInfo[] getMBeanNotifications() #String getMBeanDescription() -void setupMBeanInfo() +String getAdminDomain() +String getHostIP() +String getProcessId() </pre>

FIG 4K

42

G2Component
<pre> +G2Component(Object o) +String getAdminDomain() </pre>

FIG 4L

43

AdminComponent
<pre> +AdminComponent(Object o) +String getAdminDomain() </pre>

FIG 4M

10/48

44

CascadingAgentComponent
+String MBEANTYPE="CascadingAgent"
+String CONNECTORCLIENTCLASSNAME="com.sun.i"
+CascadingAgentComponent(CascadingAgent agent)
+String getId()
+String getType()

FIG 4N

45

HTMLAdaptorComponent
-String HTTPCONNECTIONSERVER
-HtmlAdaptorServer mHtmlAdaptor
+HtmlAdaptorComponent(HtmlAdaptorServer has)
+String getType()

FIG 4O

46

RMICConnectorServerComponent
+String RMICONNECTORSERVER="RmiConnectorServer"
+String RMICONNECTORSERVICENAMEPREFIX="Process-"
-RmiConnectorServer mTheRmiConnector=null
+RMICConnectorServerComponent(RmiConnectorServer o)
+String getRMICConnectorServiceName(int targetProcessId)
+String getType()

FIG 4P

47

ServiceComponent
-BaseService m_Service
-String MYBEANDESCRIPTION="A Service"
+ServiceComponent(Object s)
+String eType()
+String getNode()
#String getMBeanDescription()
#ModelMBeanAttributeInfo[] getMBeanAttributes()
#ModelMBeanOperationInfo[] getMBeanOperation

FIG 4Q

11/48

48

NodeComponent
<pre> -String MYMBEANDESCRIPTION="A node : runs a set" -MasterNode mNode </pre>
<pre> +NodeComponent(Object n) +String getType() #String getMBeanDescription() #ModelMBeanAttributeInfo[] getMBeanAttributes() #ModelMBeanOperationInfo[] getMBeanOperations() </pre>

FIG 4R

49

NodeAdmin
<pre> +Node createNode(NodeType type) +void startNode(NodeType nt) +void stopNode(NodeType nt) +Node getNode(NodeType type) +Integer activeNodeCount() +void recoverNodes() </pre>

FIG 4S

50

G2Runtime
<pre> -G2Runtime mInstance=new G2Runtime() -int mProcessId -String mHostIdString </pre>
<pre> -G2Runtime() +void init() +ComponentRegistry getComponentRegistry() +String getHostId() +int getProcessId() </pre>

FIG 4T

12/48

51

ComponentAddress
<pre> -Hashtable mAddressAttributes -String HOST=G2Runtime.getHostIp() -String PROCESS=""+G2Runtime.getProcessId() -AddressAttribute[] mAttributeList=new AddressAttribute[] </pre>
<pre> +ComponentAddress newAddress() +ComponentAddress getAddress(Component c) -ComponentAddress(Hashtable props) -ComponentAddress() +Hashtable getAttributes() +void setAttribute(AddressAttribute aa) +AddressAttribute getAttribute(String name) +boolean isComplete() +boolean equals(Object o) +String toString() </pre>

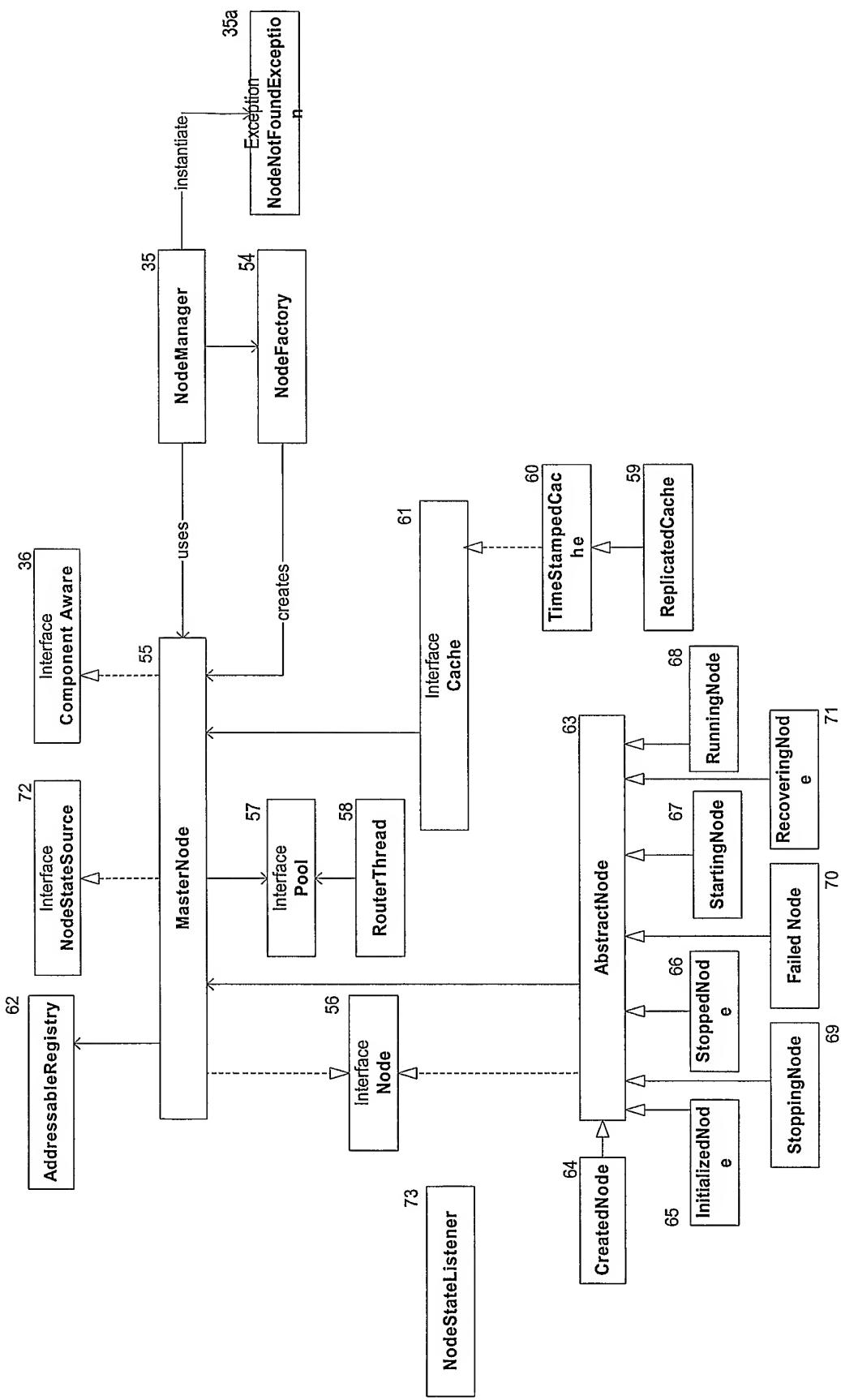
FIG 4U

52

MBeanConfig
<pre> +int RMIPORT=1099 +int HTTPPORT=8080 +String G2DOMAQIN="G2Domain" +String DEFAULTDOMAIN="DefaultDomain" +String PIDPROPERTYNAME="g2.processid" </pre>

FIG 4V

13/48

**FIG 5**

55

ComponentAware ...platform.node.MasterNode
-Thread mEventNotifier
-NodeType type
-String subType
-HashMap mStateTransitions
-HashMap mServices
-HashMap mListeners
-boolean initialised
-boolean removed
-String lastError=NOERROR
<u>-String NOERROR="NONE"</u>
-Node mState
-Pool mRouterThreadPool
-Cache mCache
-AddressableRegistry mAddressable
-void notifyListeners()
-void setType(NodeType type)
-void startServices()
-void stopServices()
+String getLastError()
+NodeType getType()
+Node getState()
+String getStateDescription()
-void initRouterThreadPool()
#void addService(HandlerContext s)
#void removeServices()
+ MasterNode()
+void addListener()
+void removeListener()
+void start()
+void stop()
+void setSubType(String st)
+String getSubType()
+void init()
+void destroy()
+Integer getAvailableRouterThread()
+RouterThread getRouterThread()
+void releaseRouterThread(RouterThread)
+Class getComponentClass()
+void recover()
#void setState(Node newState)

FIG 5A

54

```
com.genient.platform.node.NodeFactory
-Object[] NODEDEFS=( { NodeType.A
-NodeFactory instance=new NodeFactory();
# NodeFactory()
+NodeFactory getInstance()
+Node[] createNodes(NodeType[] nodetype)
+Node createNode(NodeType nodetype)
-void loadNodeDefs()
-LinkedList getApplicationAdapterServiceList()
-LinkedList getClientAdapterServiceList()
-LinkedList getBPStoreServiceList()
-LinkedList getITMServiceList()
-LinkedList getSystemManagerServiceList()
-LinkedList getDebugNodeServiceList()
```

interface ...platform.node.Node
+void init() +void start() +void stop() +void destroy() +void recover()

FIG 5C

56

interface com.genient.platform.util.Pool
+void releaseObject(Object o) +Object getObject() +int getCount() +int getSize()

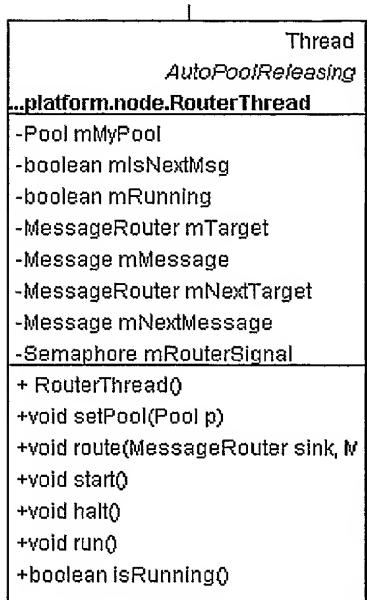
FIG 5D

57

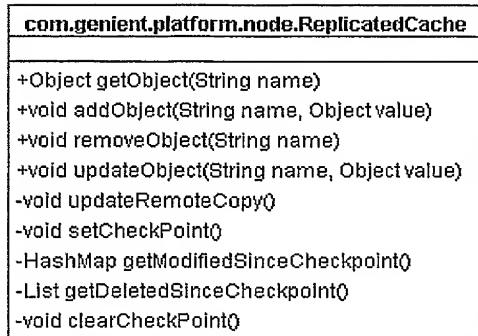
FIG 5B

15/48

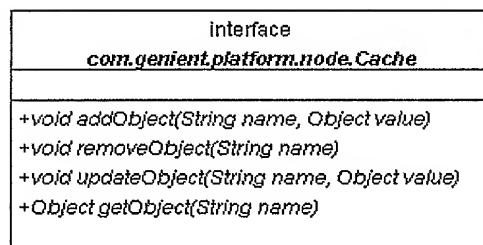
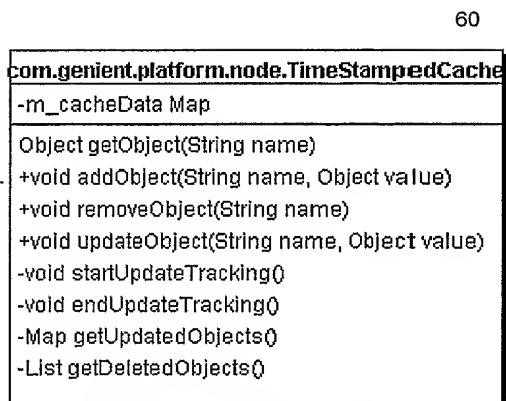
58

**FIG 5E**

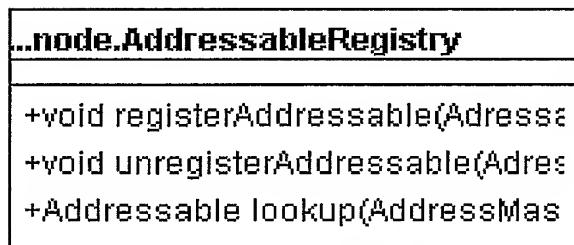
61

**FIG 5F**

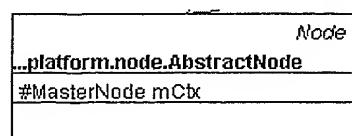
62

**FIG 5H****FIG 5G**

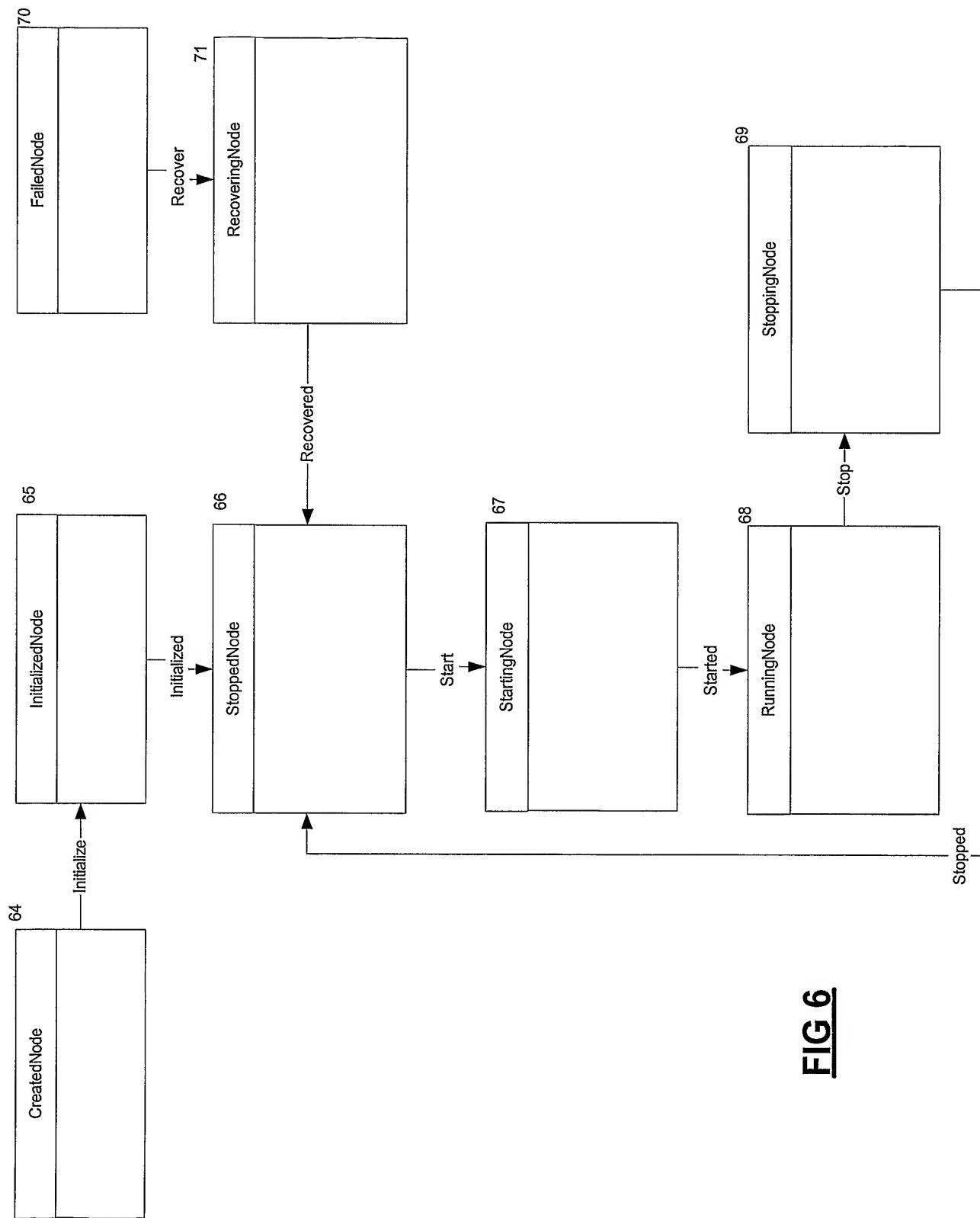
63

**FIG 5I**

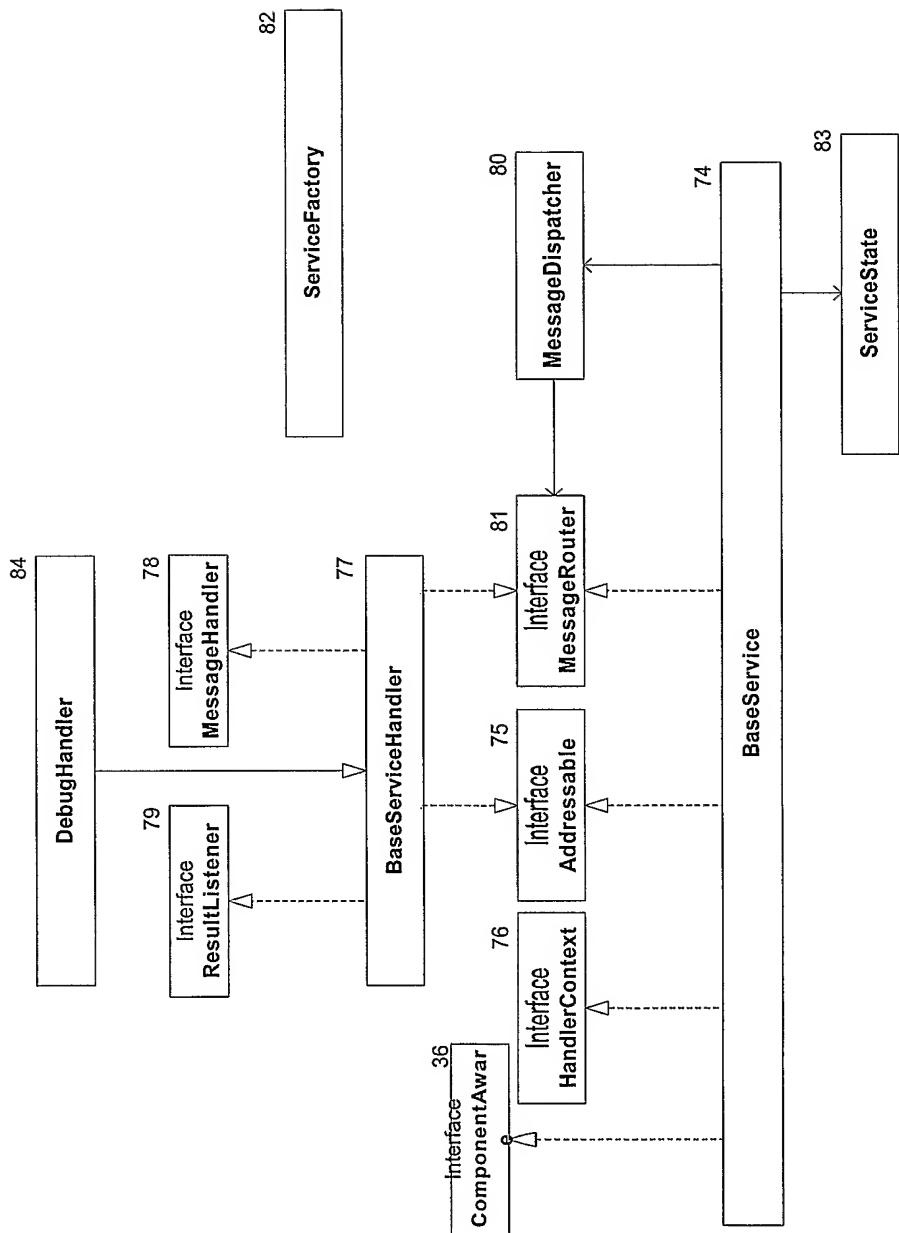
77

**FIG 5J**

16/48

**FIG 6**

17/48

**FIG 7**

18/48

```

ComponentAware
...platform.framework.BaseService
- int mMessageSendCount
- int mMessageReceivedCount
- float mMsgAvg
- HashMap mMessageListeners
- int mResultsHandled
- int mResultsMissed
- MasterNode mNode
- ServiceType mType
- ServiceState mState
- Pool mServiceableHandlerPool
- PriorityQueue mInQueue
- PriorityQueue mOutQueue
- MessageDispatcher mMyInDispatcher
- MessageDispatcher mMyOutDispatcher
- ComponentAddress mMyAddress
- AddressableRegistry mRegistry
<<constructor>>
+ BaseService(ServiceType service)
+ void initService(String host, String port)
+ ServiceType getType()
#void sendServiceMessage(Message msg)
+ void route(Message msg)
+ void updateMessageStatistics(Message msg)
+ ComponentAddress getAddress()
#void initHandlerPool(Vector handlers)
+ void startService()
+ void stopService()
+ MasterNode getNode()
+ Integer getState()
+ Integer getInQueueSize()
+ Integer getOutQueueSize()
+ Integer getMessageInCount()
+ Integer getMessageOutCount()
+ String getAverageMessageAge()
+ Integer getAvailableHandlers()
#Queue getInQueue()
#Queue getOutQueue()
+ void addResultListener(MessageId replyToId, ResultListener target)
+ ResultListener removeResultListener(MessageId replyToId)
+ Integer getResultListenersCount()
+ void receivedResult()
+ Integer getResultsHandled()
+ void missedResult()
+ Integer getResultsMissed()
+ Class getComponentClass()
-MESSAGEToHandlerForwarder
-MESSAGEToRouterForwarder

```

74

```

interface
com.genient.platform.frameworkHandlerContext

```

```

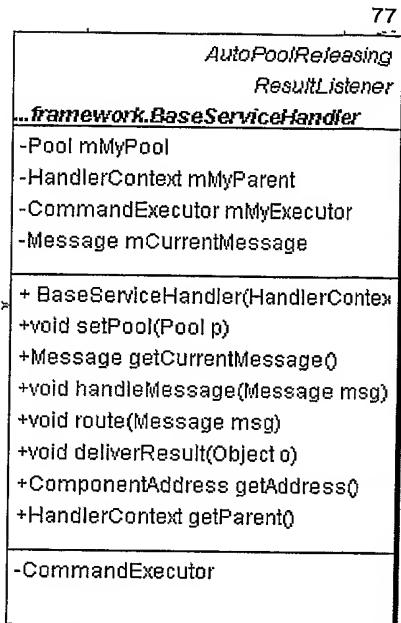
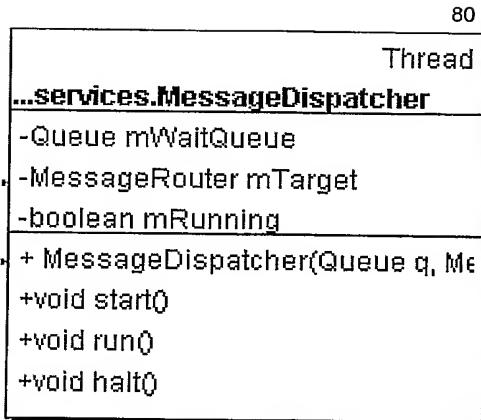
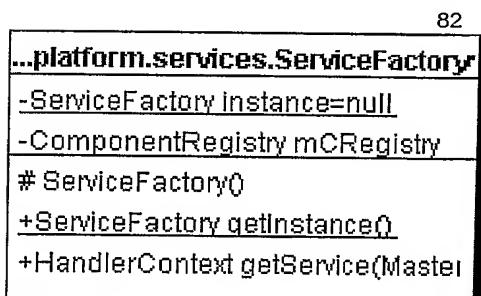
+ ServiceType getType()
+ void route(Message msg)
+ ComponentAddress getAddress()
+ Integer getState()
+ void addResultListener(MessageId replyToId, ResultListener target)
+ ResultListener removeResultListener(MessageId replyToId)
+ void registerAddressable(Addressable obj)
+ void unregisterAddressable(Addressable obj)
+ ArrayList lookupAddressables(AddressMask mask)
+ void addResultListener(MessageId replyToId, ResultListener target)
+ void addPersistentResultListener(MessageId replyToId, ResultListener target)

```

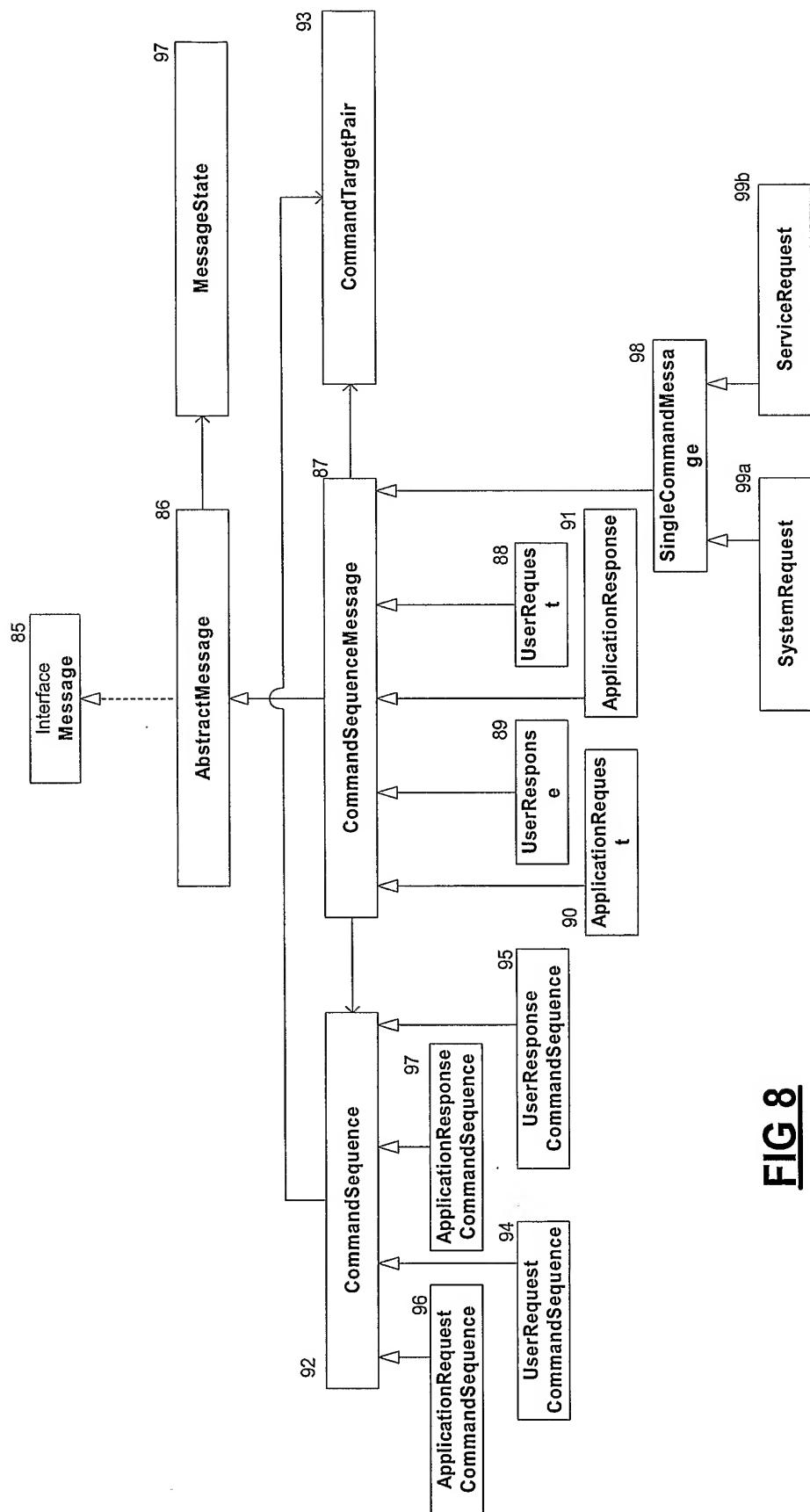
76

FIG 7B**FIG 7A**

19/48

**FIG 7C****FIG 7D****FIG 7E****FIG 7F**

20/48

**FIG 8**

86

<pre>AbstractMessage</pre>	<pre> -mClonedFrom : MessageId -mCreateTime : long -mHeaderMap : HashMap -mHeaders : HashMap -mHistory : ArrayList -mid : MessageId -mNVPs : HashMap -mSource : AddressMask -mState : MessageState </pre>
<pre>Interface Message</pre>	<pre> +void addMessageNVP(name:String, value:Serializable) +void cancel() +void commandFailed (e:RequestProcessingException) +void commandSucceeded() +Message copyWithNewId() +AddressMask getAddressMask() +long getAge() +Command getCommand() +GenientID getContext() +Object getHeaderValue (headername : String) +Iterator getHistory() +MessageId getId() +Serializable getMessageNVPValue (name : String) +MessageId getRootId() +MessageType getType() +boolean hasFinished() +Map headersAsMap() +boolean isMultiDelivery() +boolean isValidHeaderName (name: String) +leaveProcess() +void readObject (stream: ObjectInputStream) +void removeMessageNVP (name: String) +void setHeaderValue (headername: String, headervalue:Object) </pre>

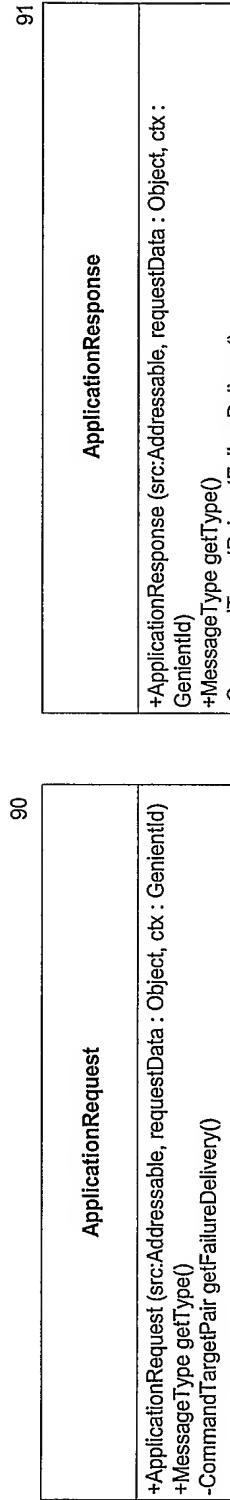
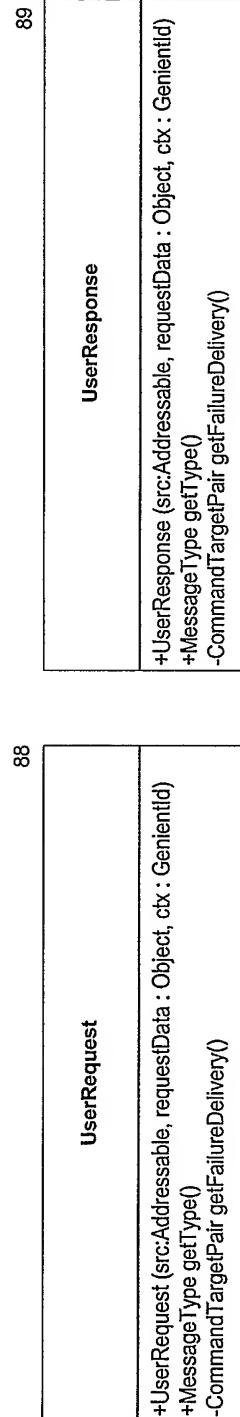
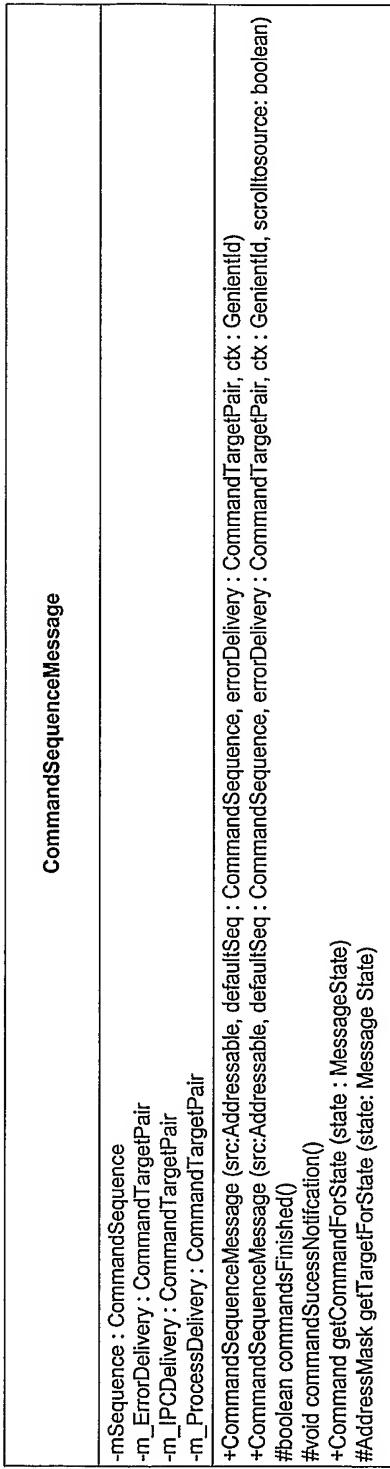
85

<pre>Interface Message</pre>	<pre> +void addMessageNVP(name:String, value:Serializable) +void cancel() +void commandFailed (e:RequestProcessingException) +void commandSucceeded() +Message copyWithNewId() +AddressMask getAddressMask() +long getAge() +Command getCommand() +GenientID getContext() +Object getHeaderValue (headername : String) +Iterator getHistory() +MessageId getId() +Serializable getMessageNVPValue (name : String) +MessageId getRootId() +MessageType getType() +boolean hasFinished() +Map headersAsMap() +boolean isMultiDelivery() +boolean isValidHeaderName (name: String) +leaveProcess() +void readObject (stream: ObjectInputStream) +void removeMessageNVP (name: String) +void setHeaderValue (headername: String, headervalue:Object) </pre>
--	---

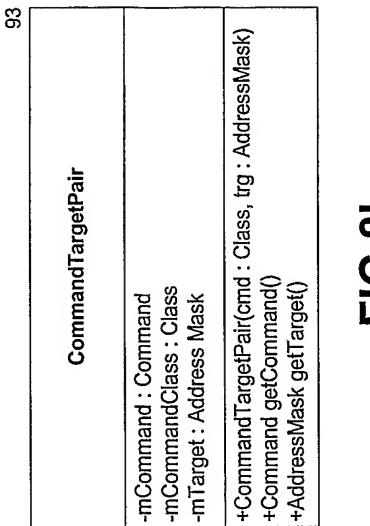
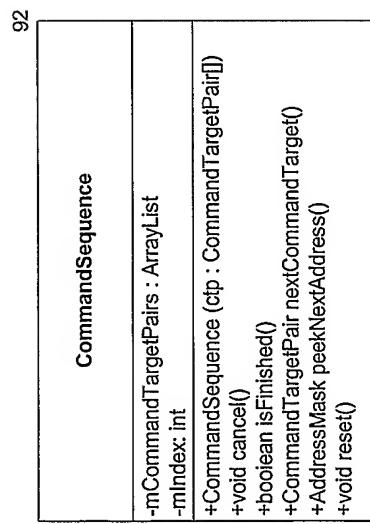
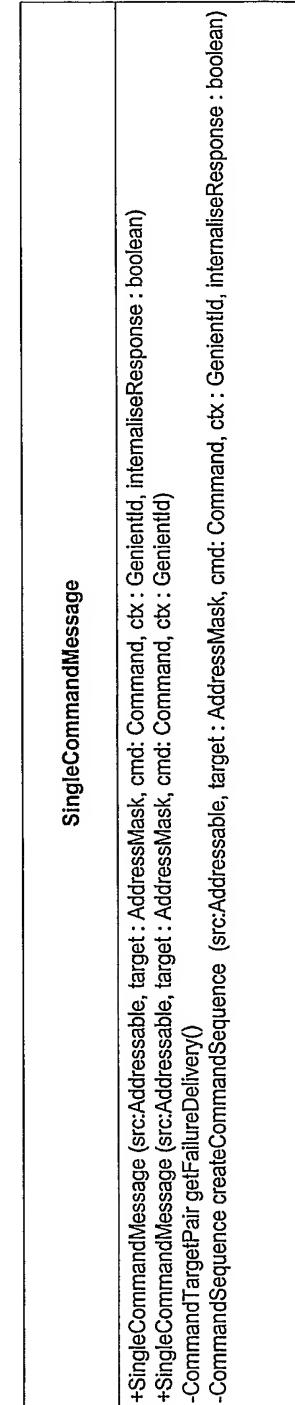
FIG 8A**FIG 8B**

22/48

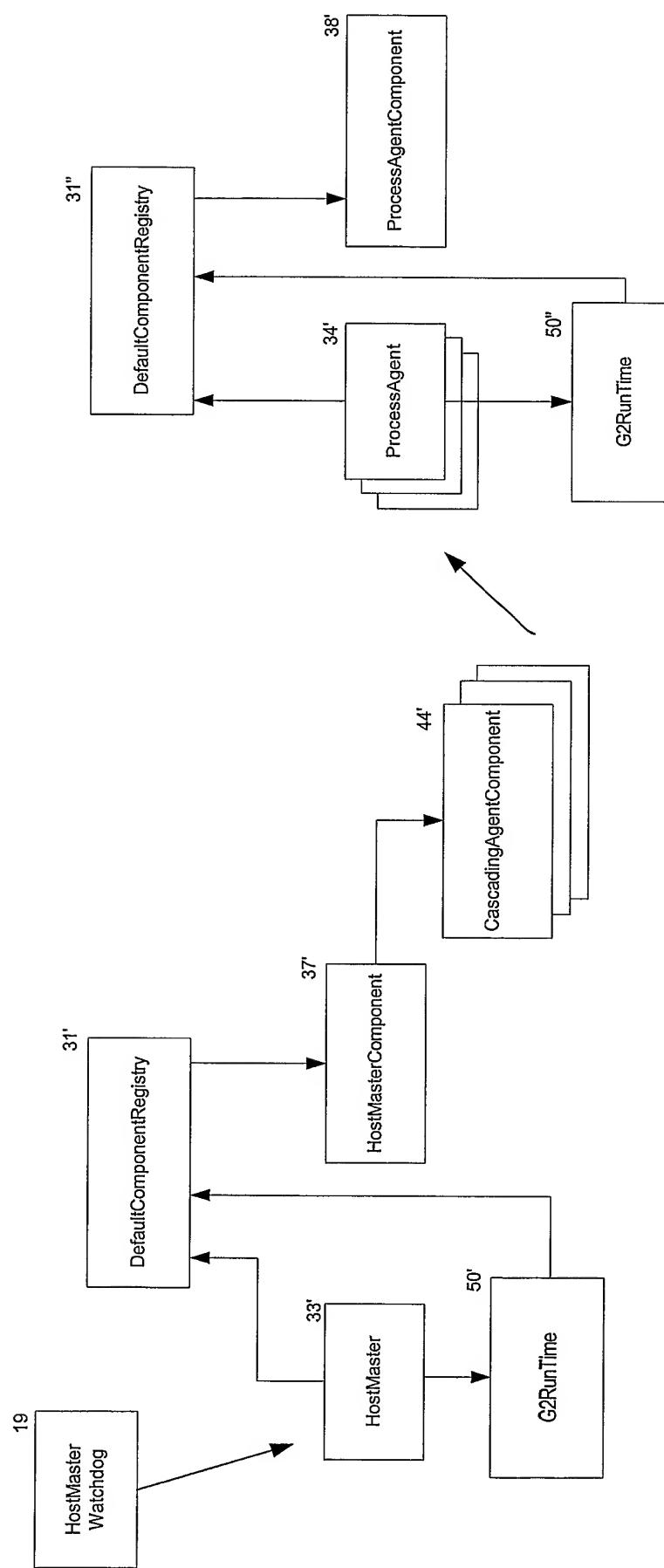
87



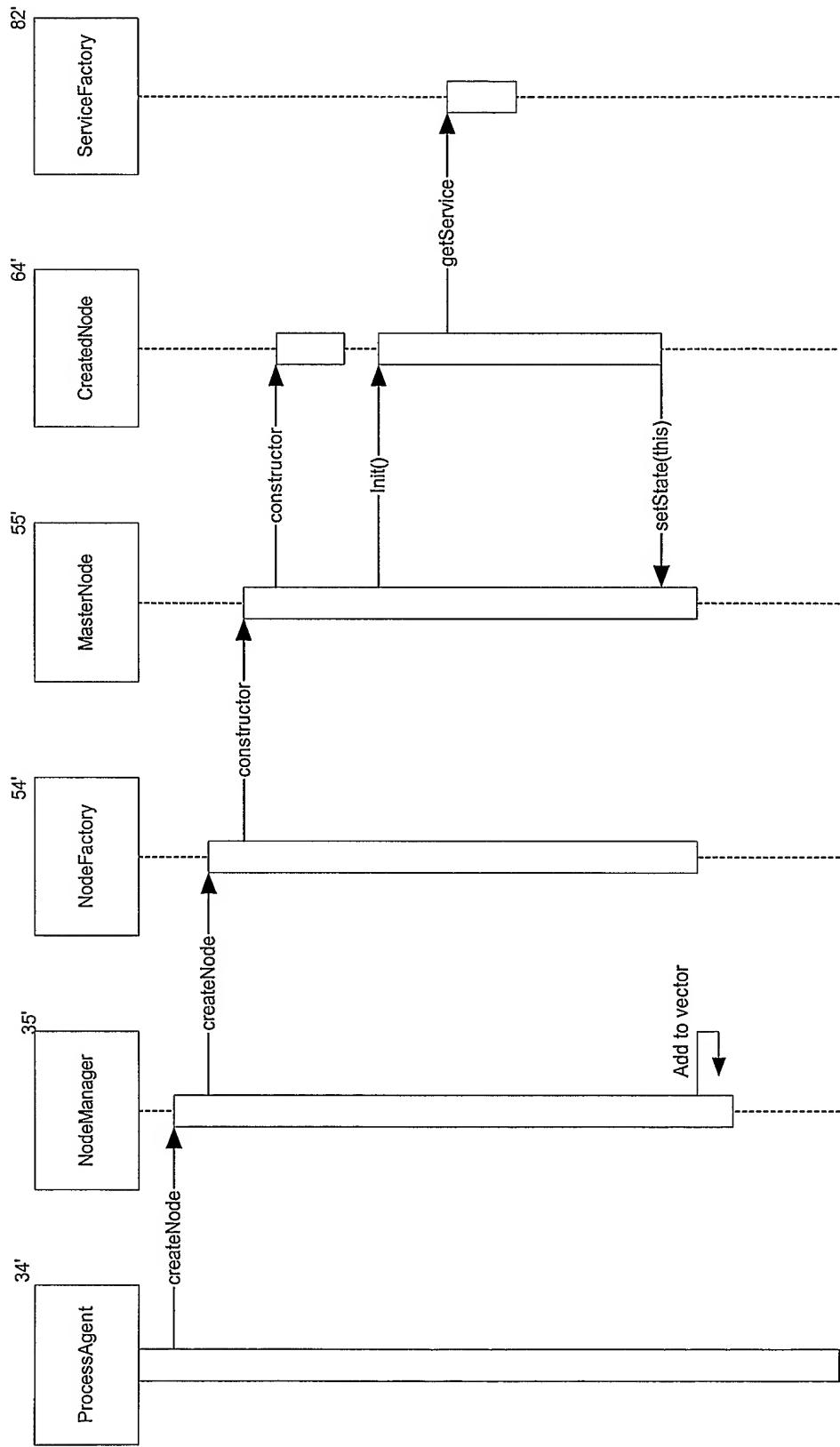
23/48

**FIG 8I****FIG 8H****FIG 8K**

24/48

**FIG 9**

25/48

**FIG 10**

26/48

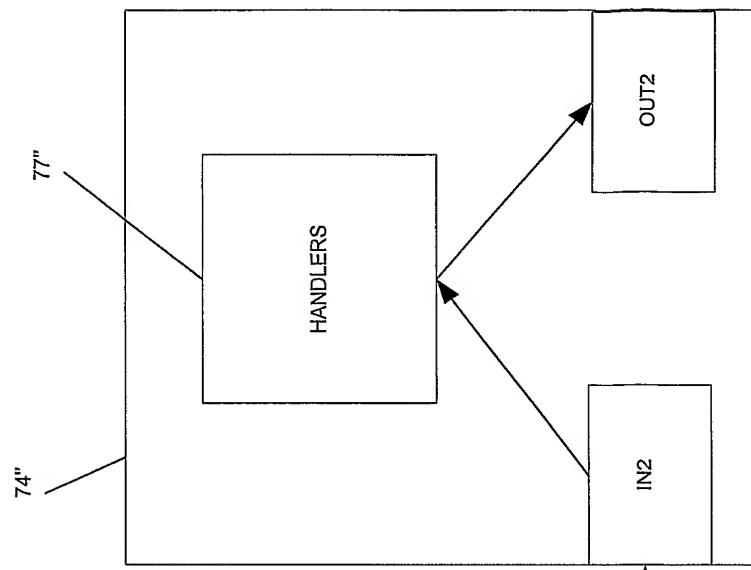
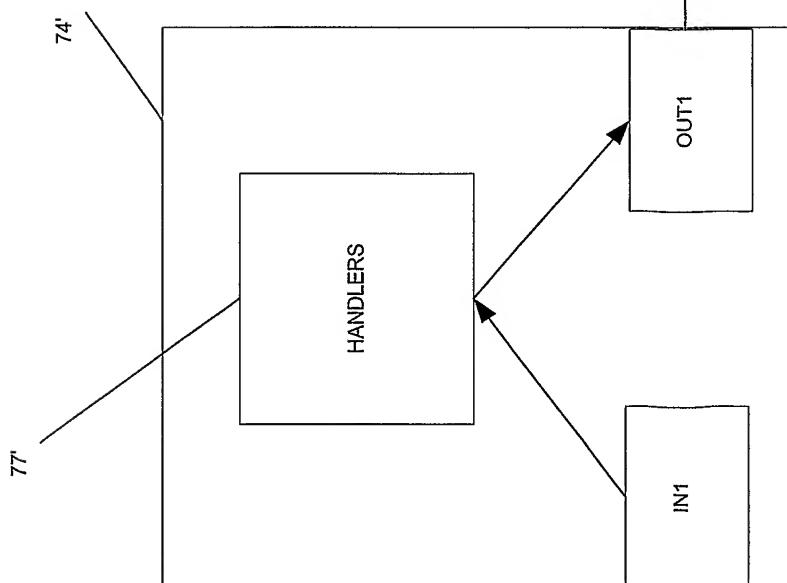
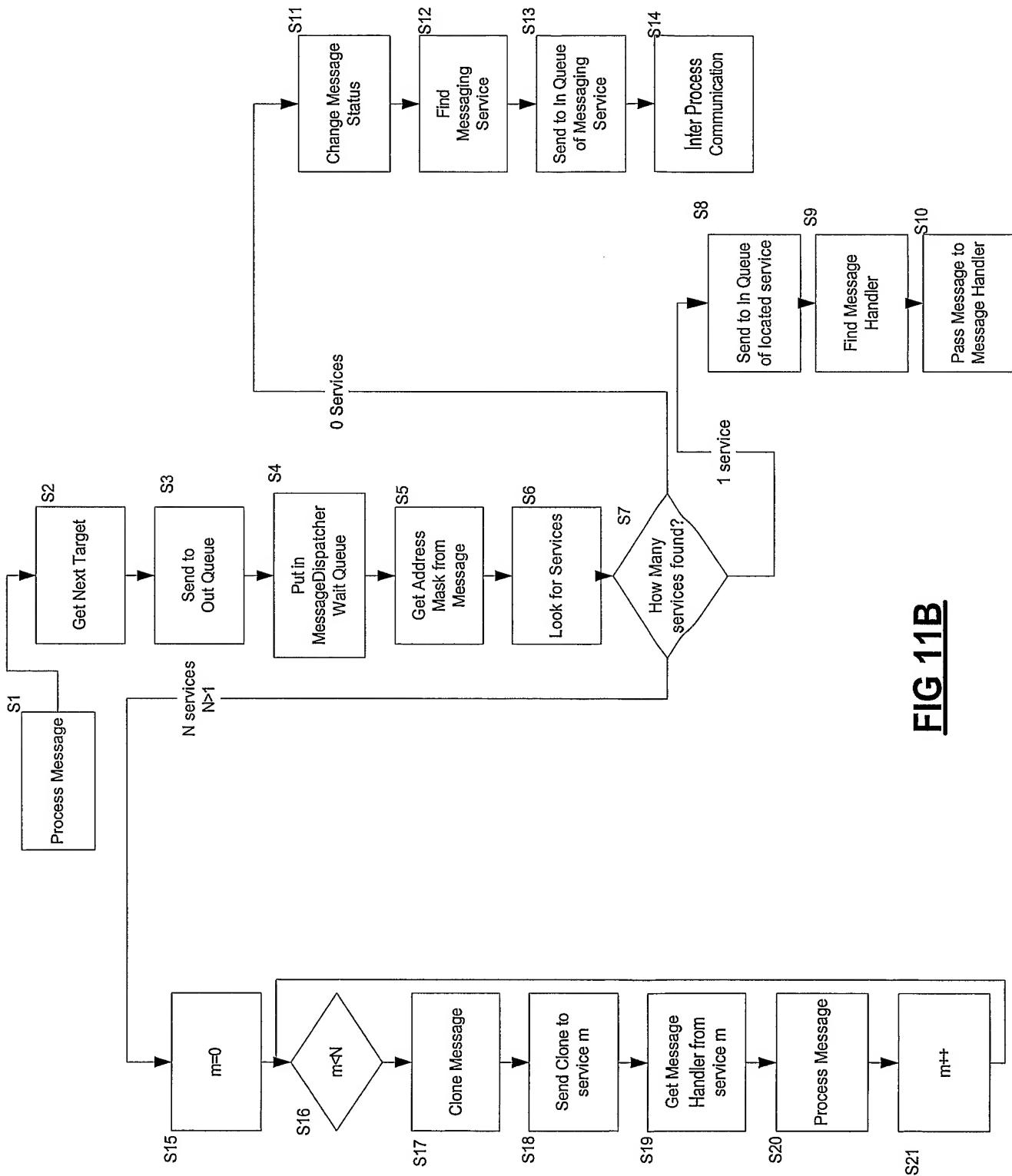
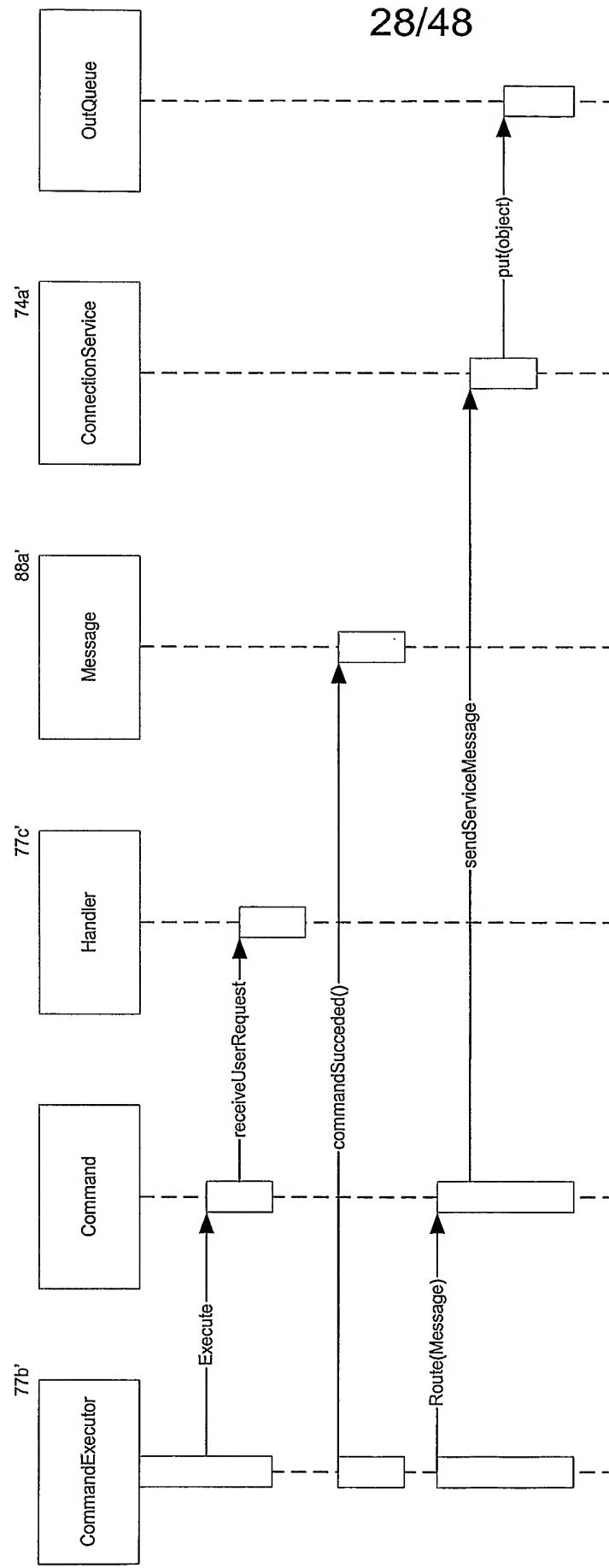


FIG 11A

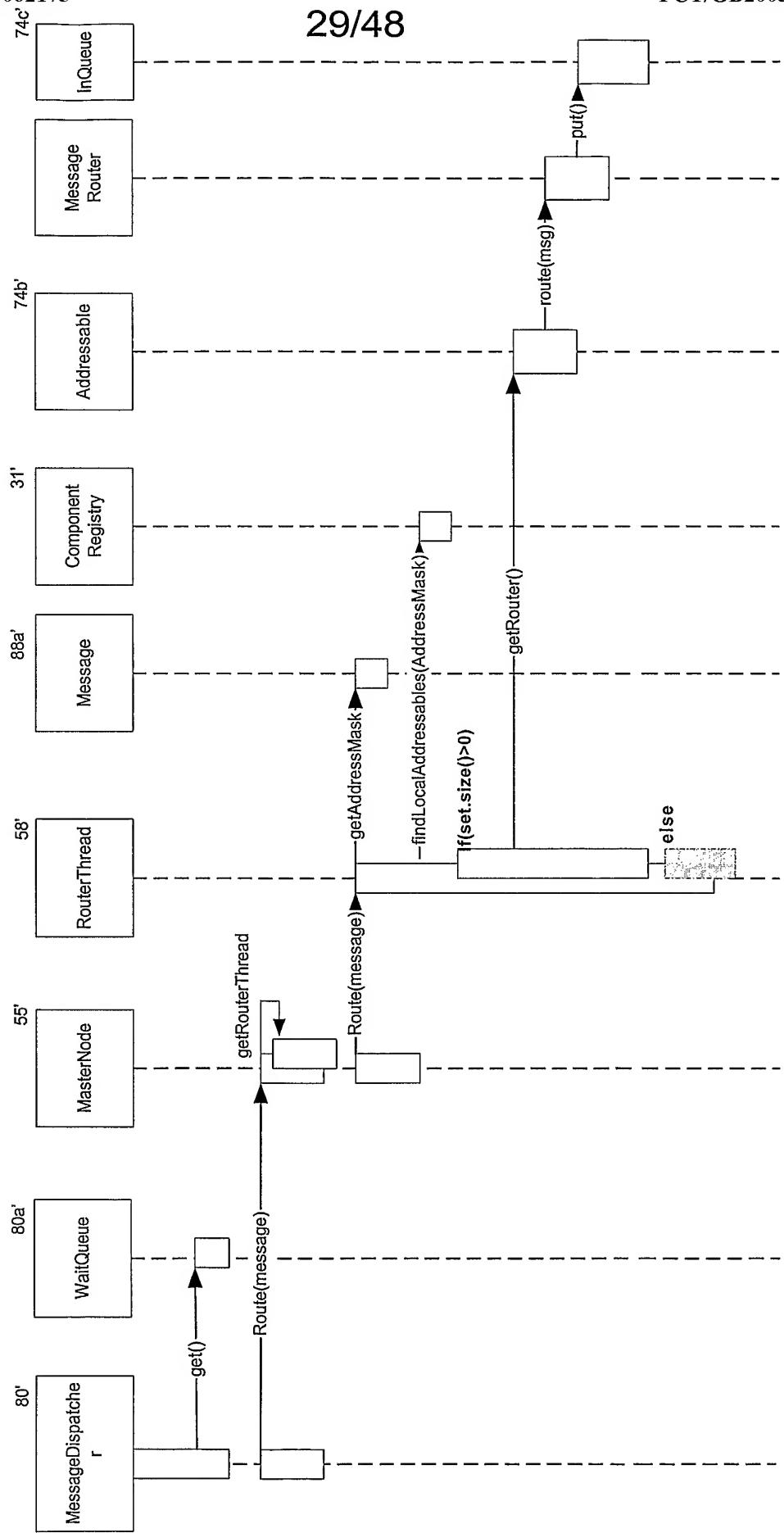


**FIG 11B**

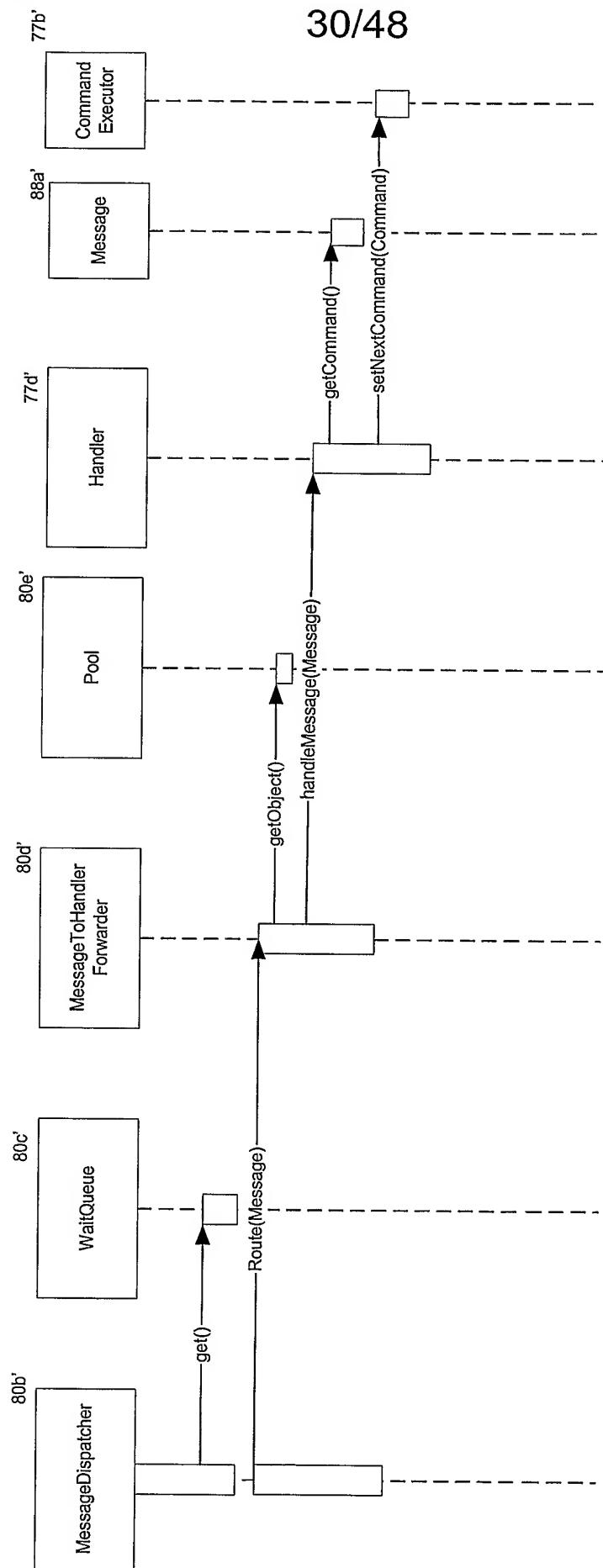
28/48

**FIG 12A**

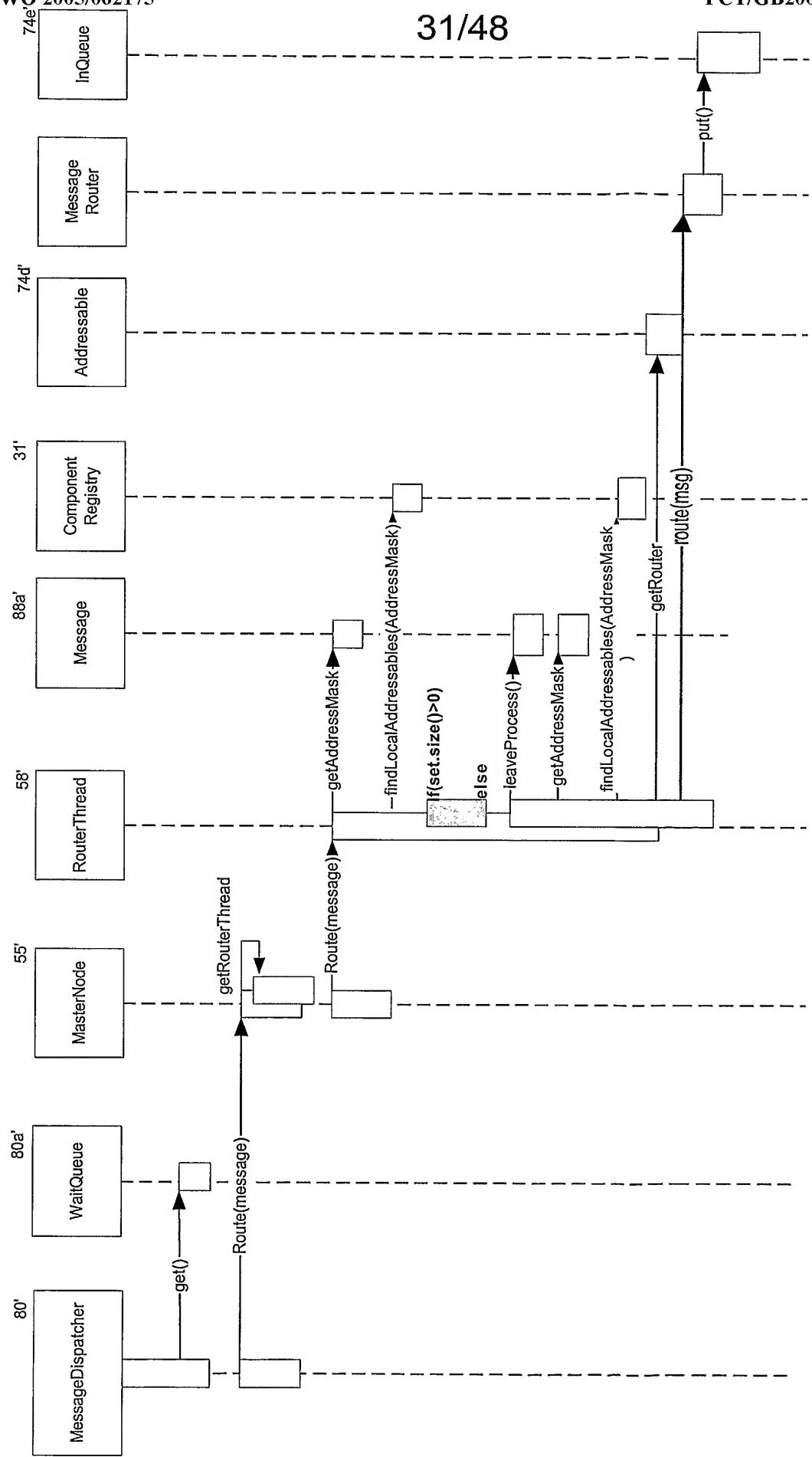
29/48

**FIG 12B**

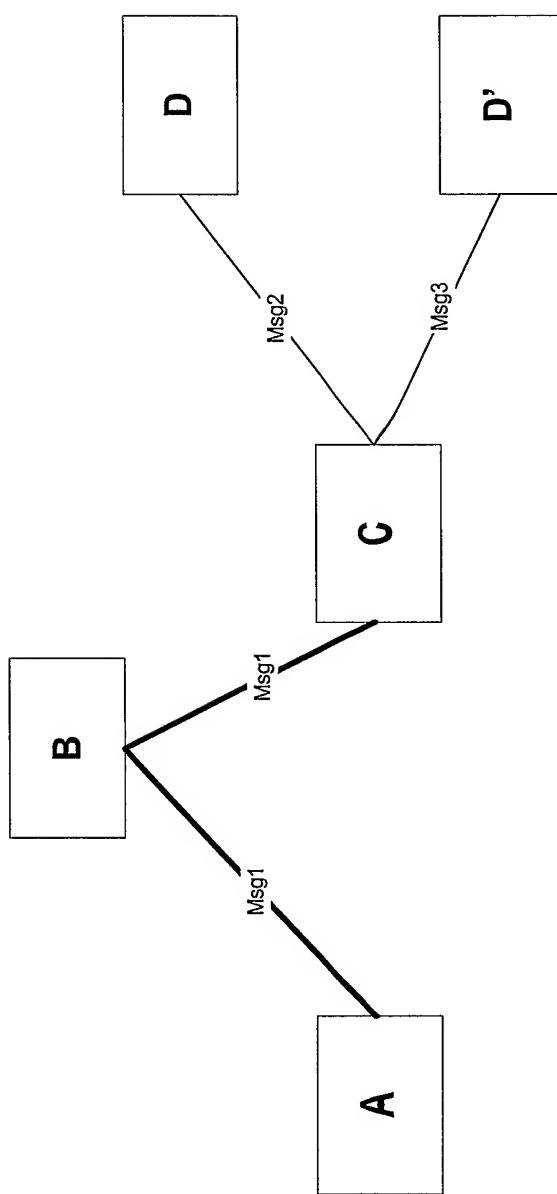
30/48

**FIG 12C**

31/48

**FIG 12D**

32/48

**FIG 13**

33/48

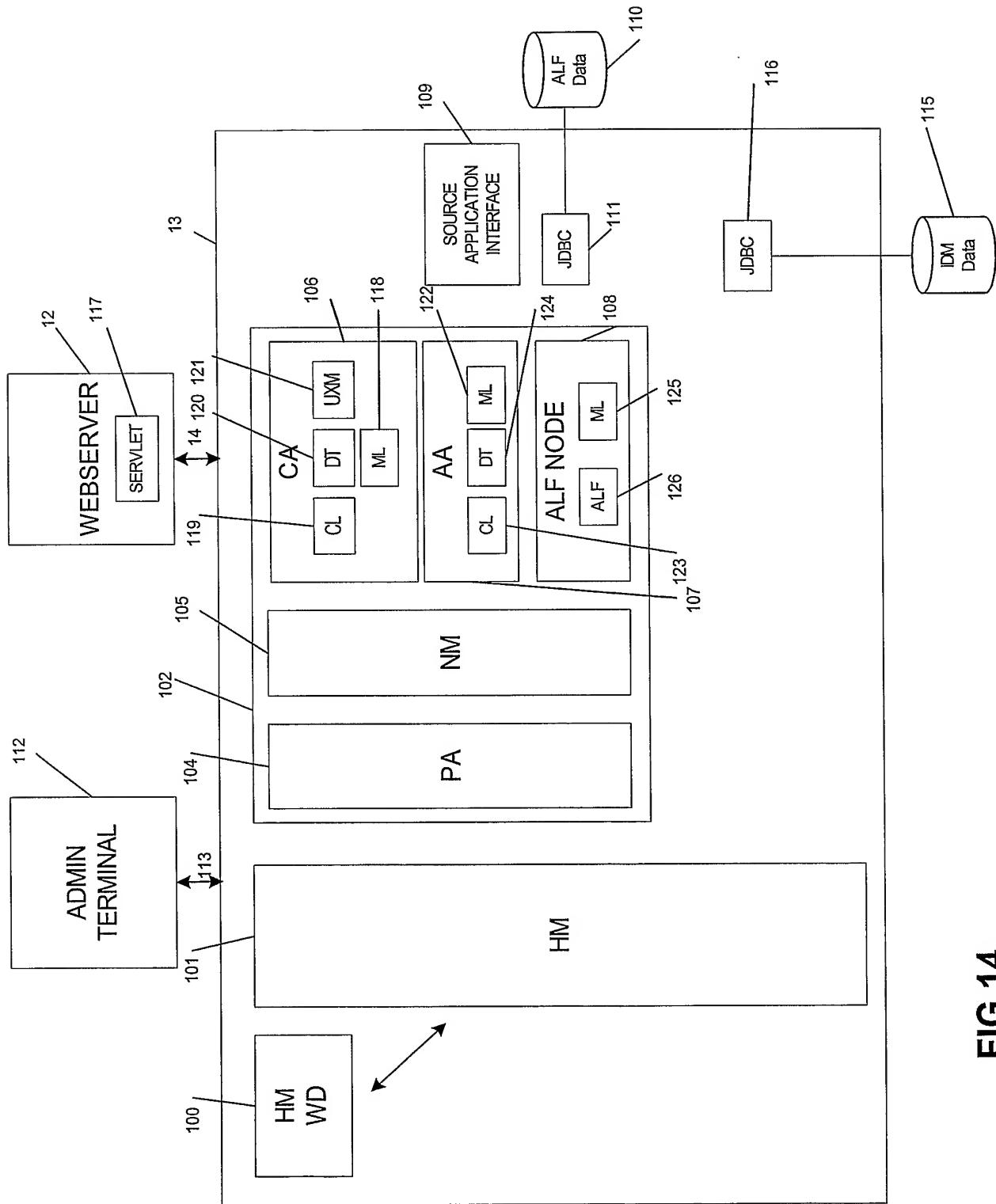
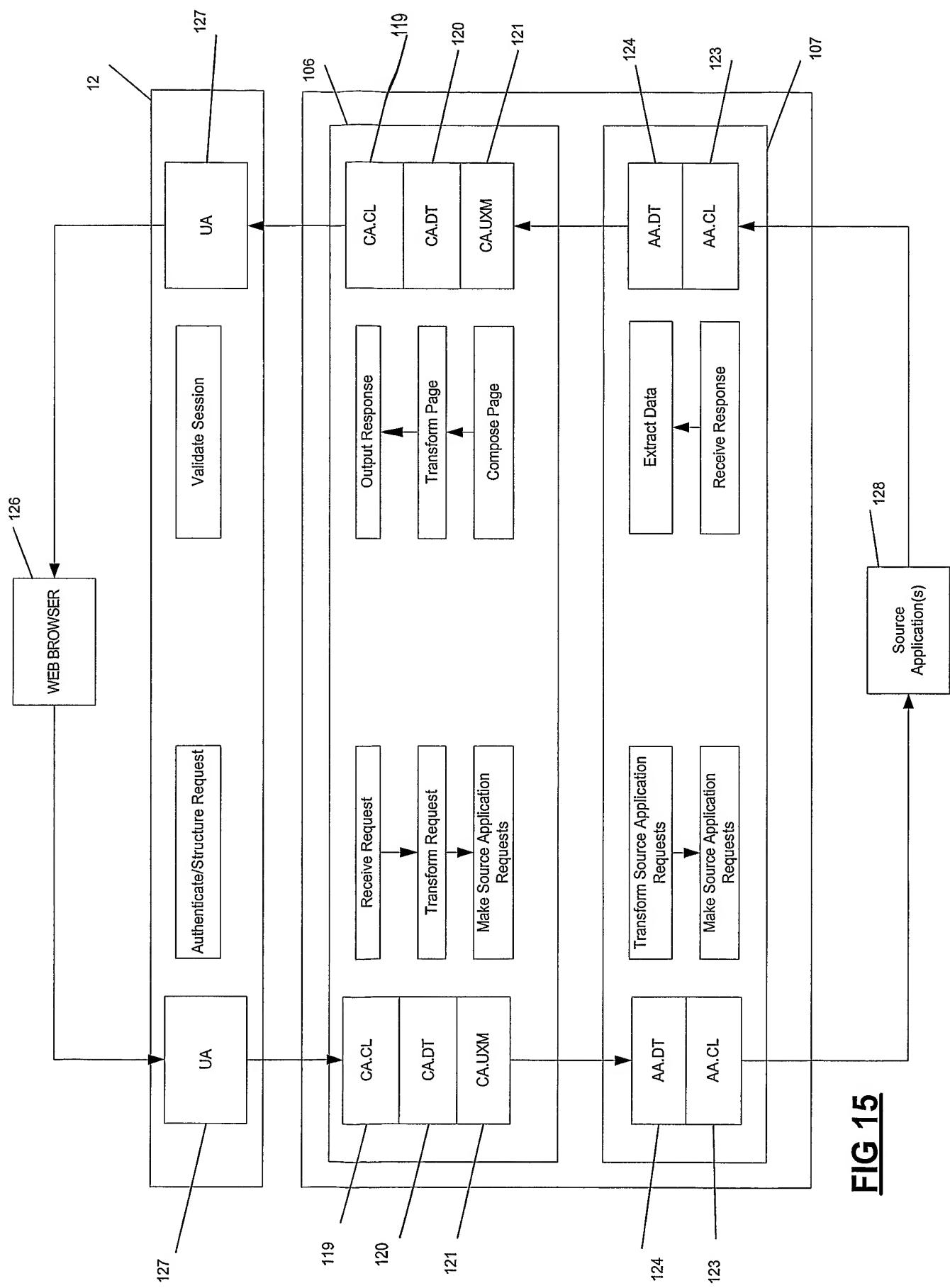


FIG 14

34/48



35/48

Parameter	Req.	Type	Values	Notes
g2jms.config	Y	Str		File containing the jms configuration properties for the Listener e.g. g2jms.config=c:\conf\jms.properties
g2listener	Y	Str		This is the name of the node under the UAConfigs node in the IDM containing the listener configuration. e.g. g2listener=weblistener
g2config.instance	Y	Str		Node in the IDM that has an instance definition for the Composite Application. [shouldn't be necessary but is in current release] e.g. g2config.instance=googleweb
g2config.home	Y	Str		Directory containing the IDM properties file for the Composer/Listener. This has to point to a directory containing the file rather than the file. e.g. g2config.home=c:\GCP\2.3.0\conf
g2tracer.conf	Y	Str		File containing the logger configuration for the Listener. e.g. g2tracer.conf=c:\conf\listenerlog.conf

FIG 16

Parameter	Req.	Type	Values	Notes
G2config.home	Y	Str		Directory containing the IDM properties file for the Composer/Listener. This has to point to a directory containing the 'IDM.properties' file rather than the file itself e.g. g2config.home=c:\GCP2.3.0\conf
g2config.instance	Y	Str		Node in the IDM that has an instance definition for the Composite Application. e.g. g2config.instance=googleweb
g2config.webhost	Y	Str		Host-name (or IP address) and port, of the Web Server used with the Listener. Used to rewrite the pages served up through the composer to point back to the listener. e.g. g2config.webhost=192.168.1.229:80
g2config.baseipport	Y	Str		Specify the Admin console port e.g. g2config.baseipport=8080
g2jms.config	Y	Str		File containing the jms configuration properties for the Composer e.g. g2jms.config=c:\conf\jms.properties
g2basetracer.conf	Y	Str		File name containing the logger configuration(s) for the Listener – it will look for a file per process so there needs to be at least a '0' and a '1' log file. e.g. g2tracer.conf=c:\conf\composerlog.conf

FIG 17

36/48

Name	Req.	Type	Values	Notes
APPLICATION_C LASS	Y	Str		Destination service for the incoming user request message that will be generated. This is the application class name used for the composite application.
UXM_MODIFIED	Y	Str	true/false	Set to true if the request is to be processed by the Composer. If false, then the UXM will not process the request and pass it on to the AA (pass through applications)
UXM_TREEID	Y	Str		The 'id' (from the control data-group) of the TreeRootNode targeted by this request.
UXM_NODEID	Y	Str		The 'id' (from the control data-group) of the TreeNode (within the hierarchy defined by UXM_TREEID) targeted by this request.

FIG 18

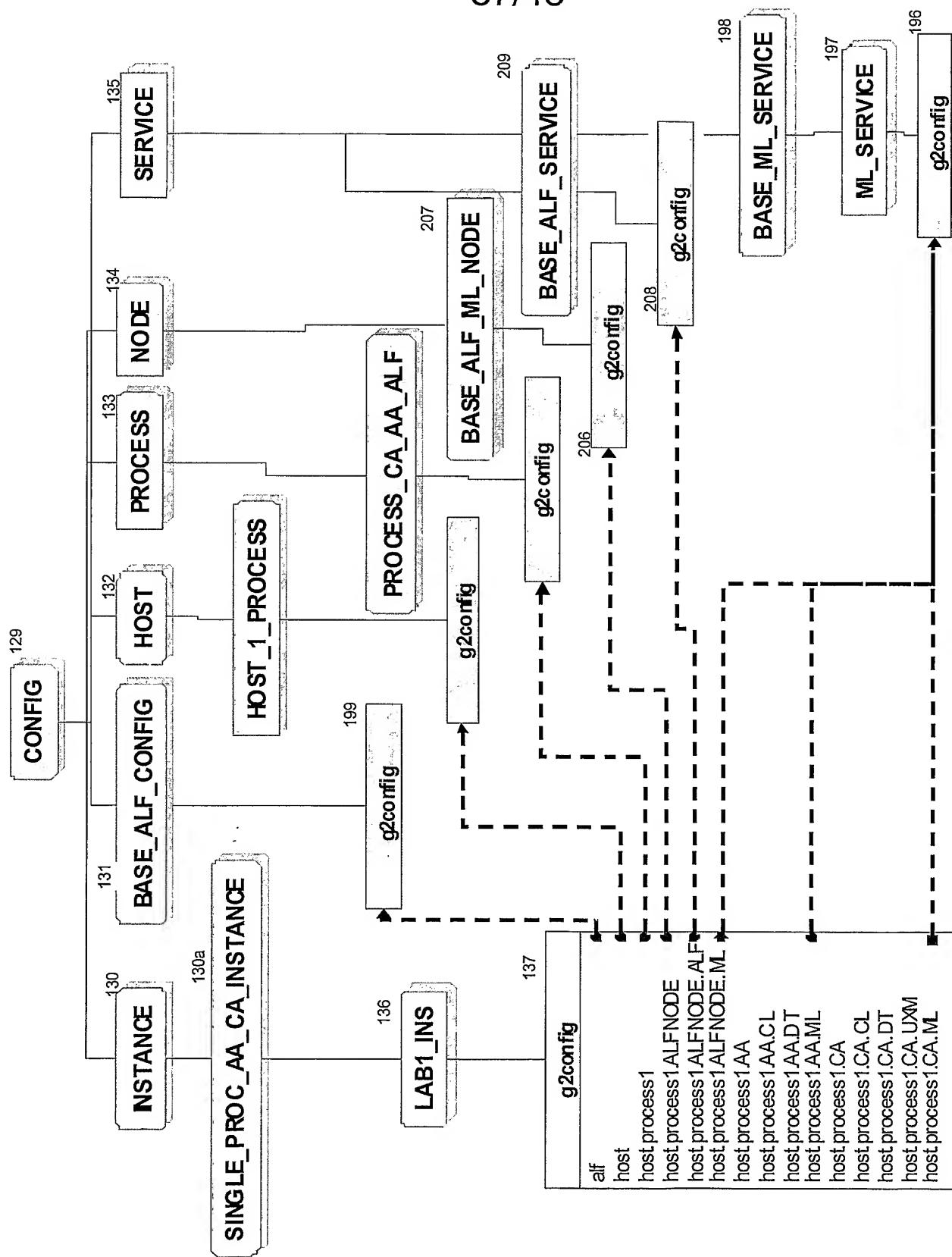
```

hm.g2config.baseipport=9080
hm.rmiport=1299
hm.g2config.webhost=10.230.226.162
hm.g2config.instance=MyInstance
hm.g2jms.config=/usr/genient/composer/conf/jms.properties
hm.g2tracer.conf=/usr/genient/composer/conf/composer-log.properties
pa.x.Xrms512m
pa.x.Xmx512m
pa.x.XXl:NewSize=64m
pa.x.XXl:MaxNewSize=64m
pa.x.XXl:+DisableExplicitGC
pa.x.g2config.webhost=10.230.226.162
pa.x.g2config.home=/usr/genient/composer/conf
pa.x.g2config.instance=MyInstance
pa.x.g2jms.config=/usr/genient/composer/conf/jms.properties
pa.x.g2tracer.conf=/usr/genient/composer/conf/composer-log.properties
pa.x.java.net.ssl.trustStore=/usr/genient/j2re1_3_1_08/lib/security/g2cert
pa.x.java.net.ssl.keyStoreType=pkcs12
pa.x.java.net.ssl.keyStorePassword=thgirwtfos
pa.x.java.net.ssl.keyStore=/usr/genient/j2re1_3_1_08/lib/security/xF8FF_2_XEG000004.p12
pa.x.java.protocol.handler.pkgs=HTTPClient

```

FIG 19

37/48

**FIG 20**

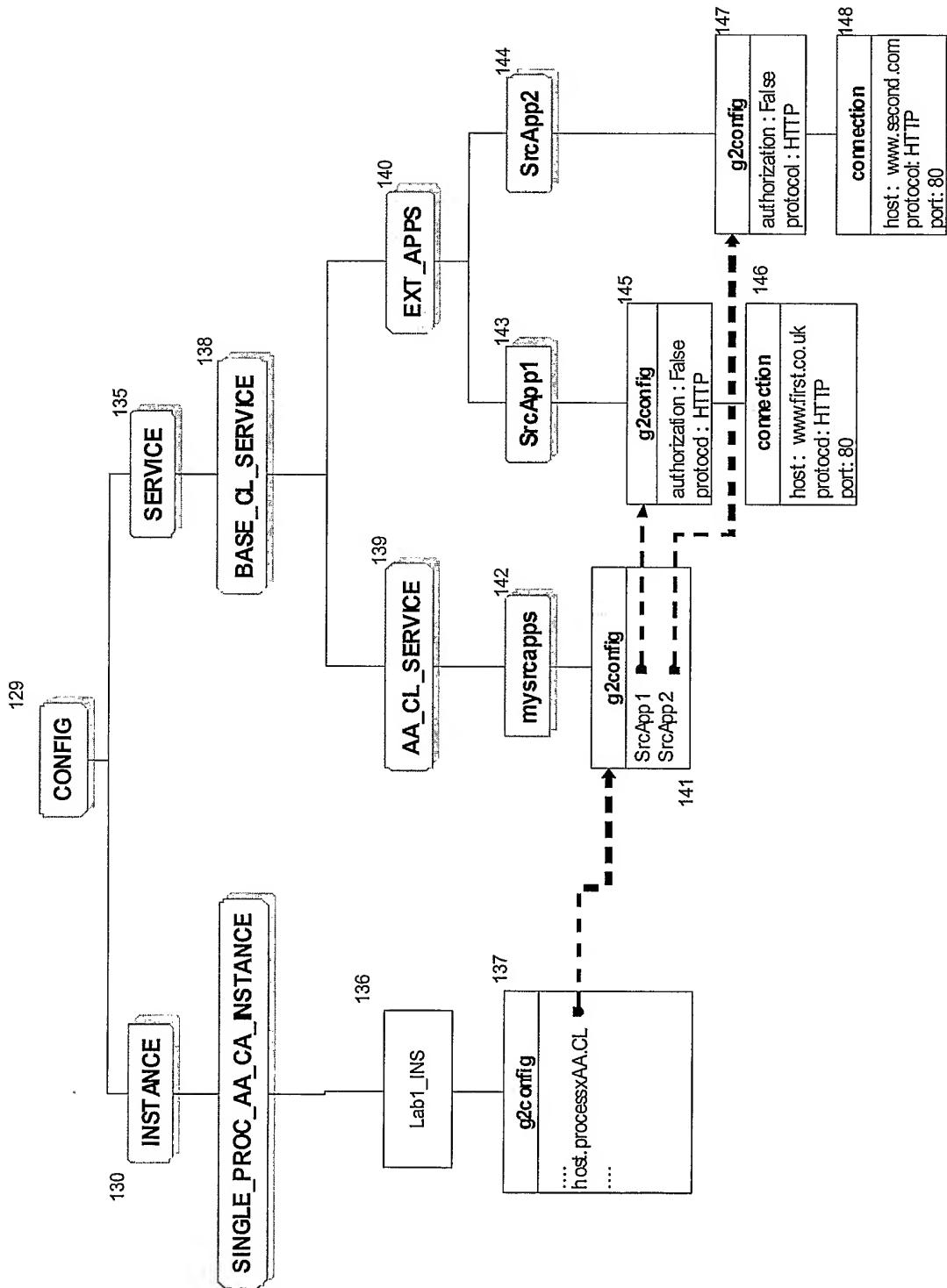
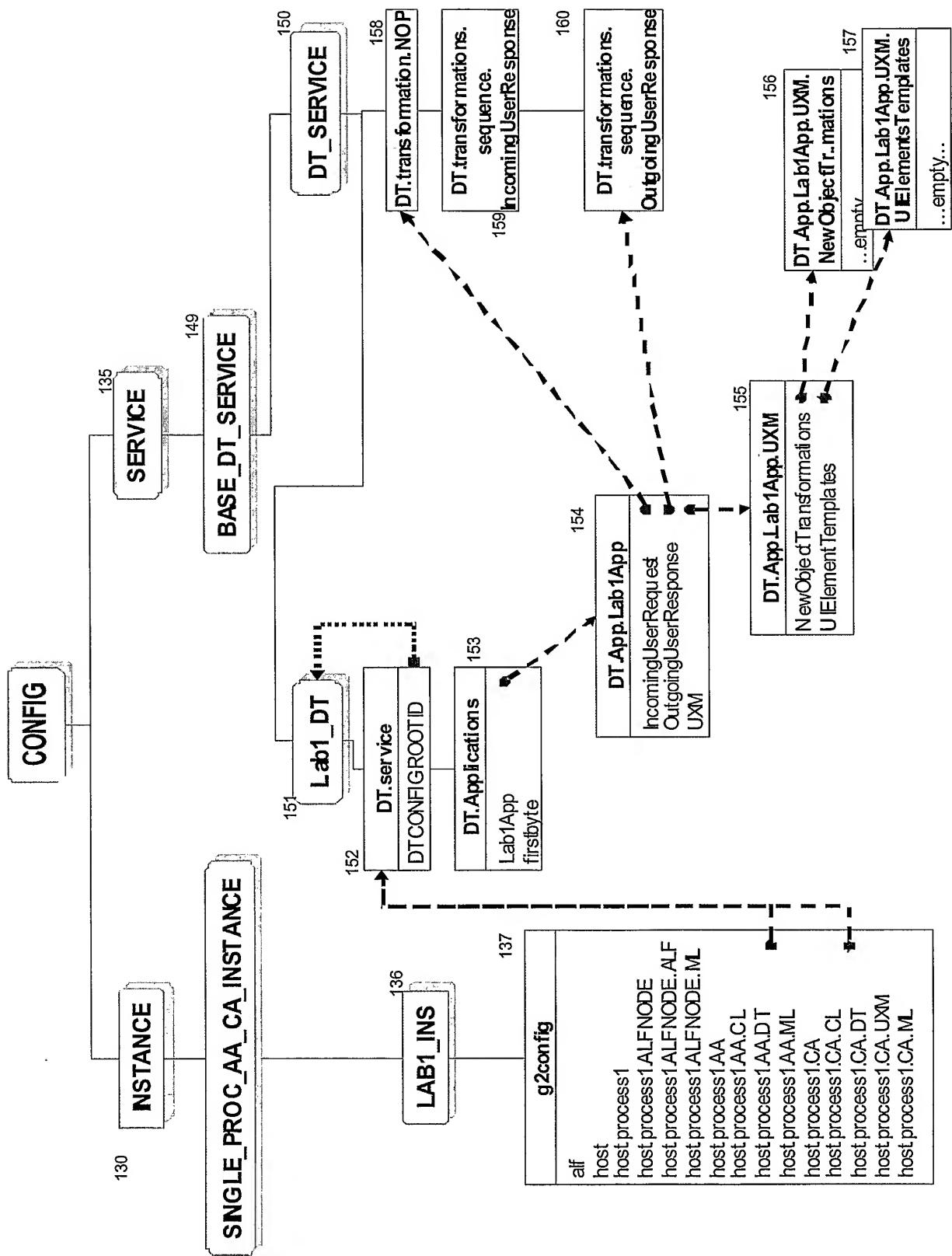
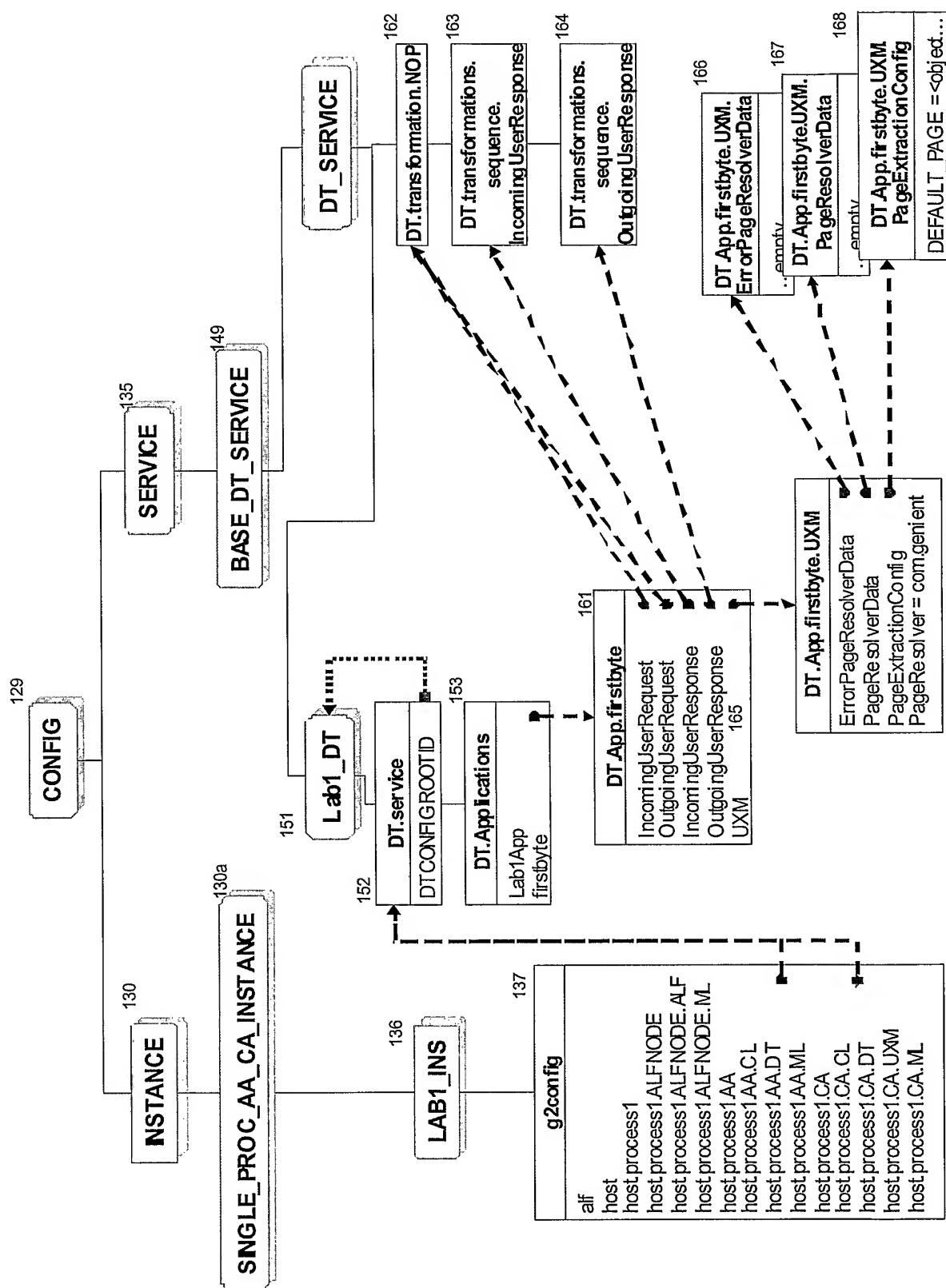


FIG 21

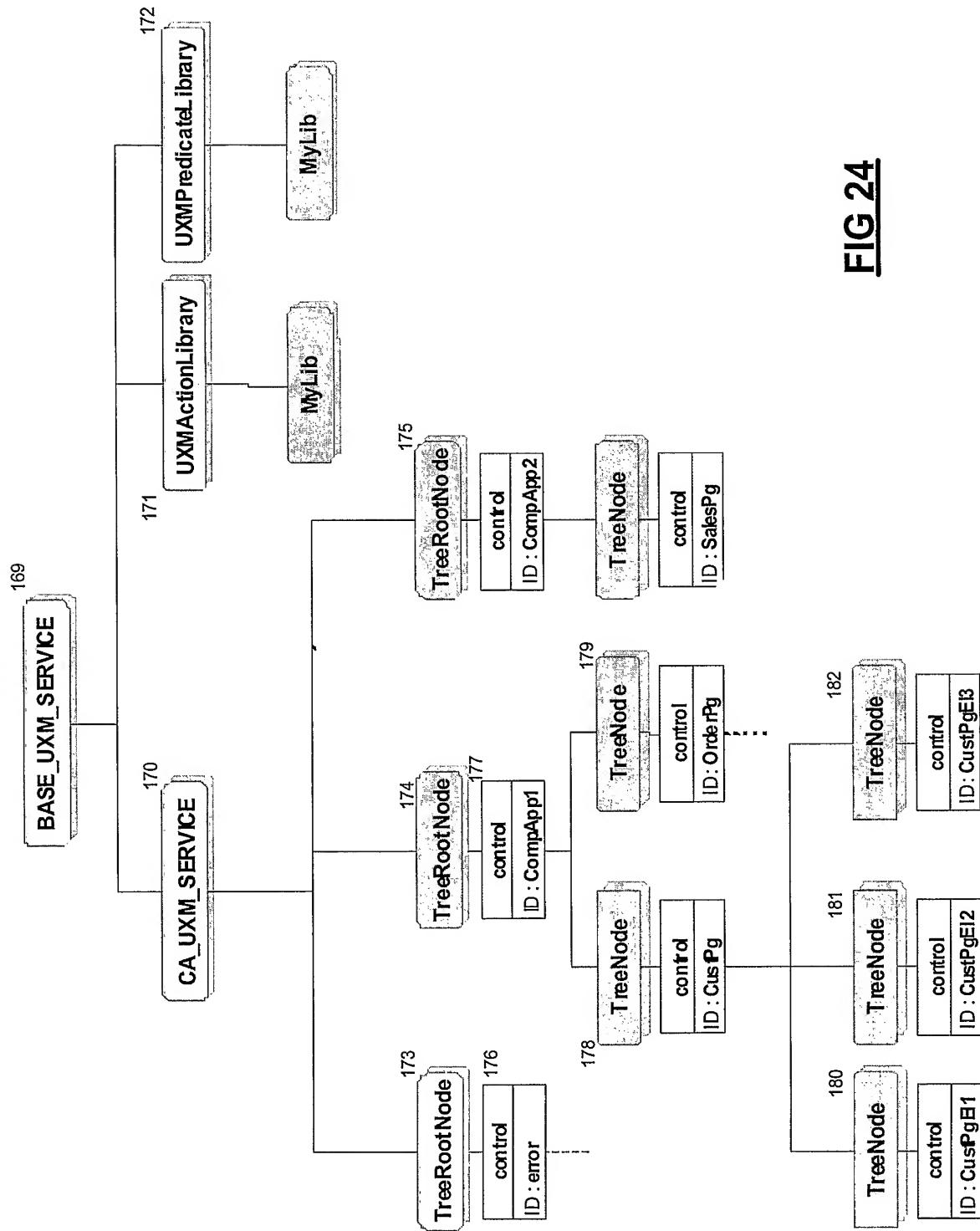
39/48

**FIG 22**

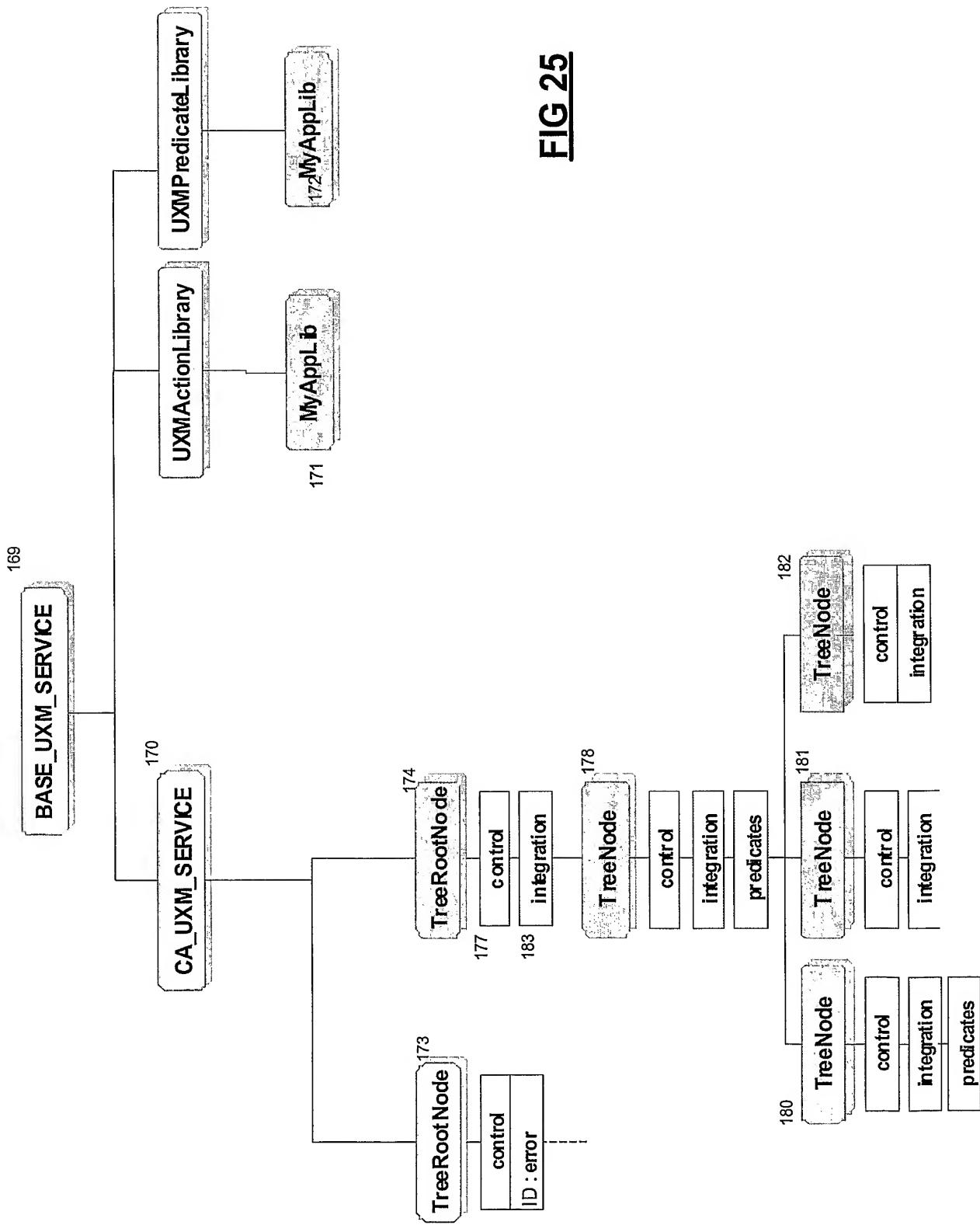
40/48

**FIG 23**

41/48

**FIG 24**

42/48

FIG 25

43/48

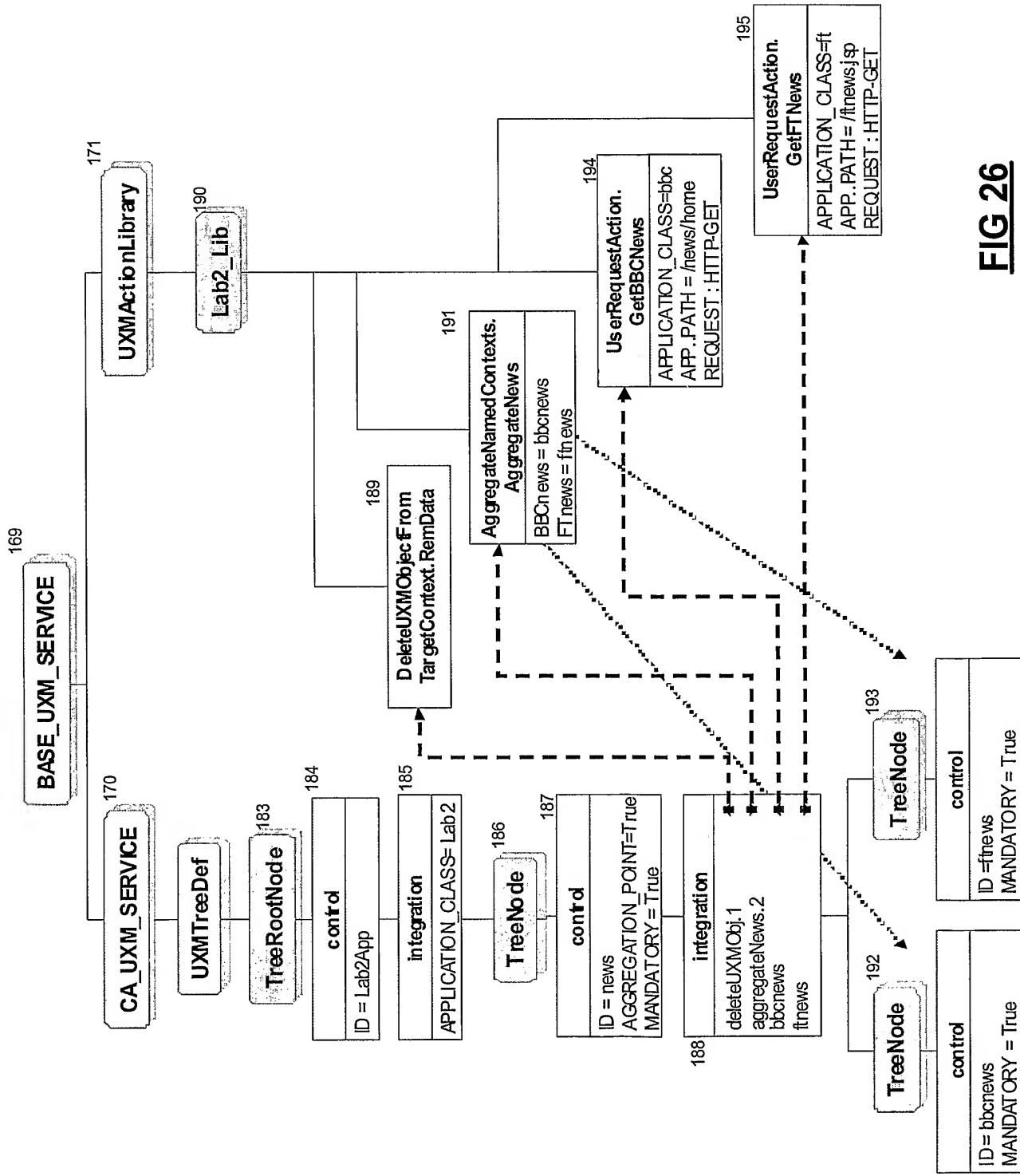
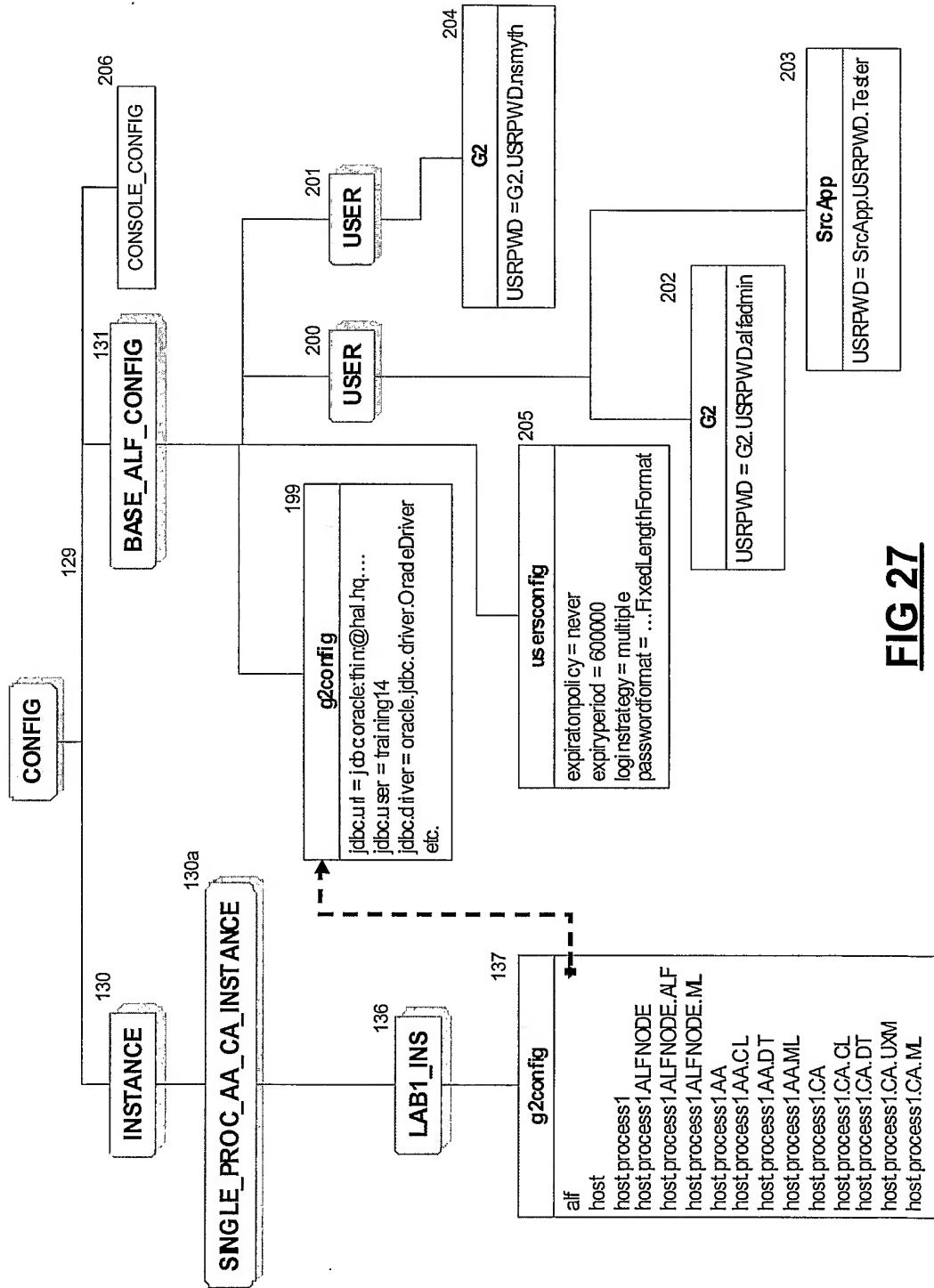
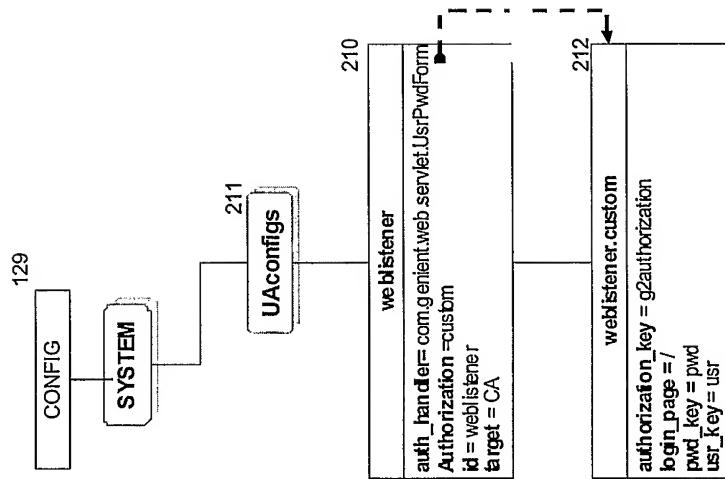


FIG 26

44/48

**FIG 27**

45/48

**FIG 28**

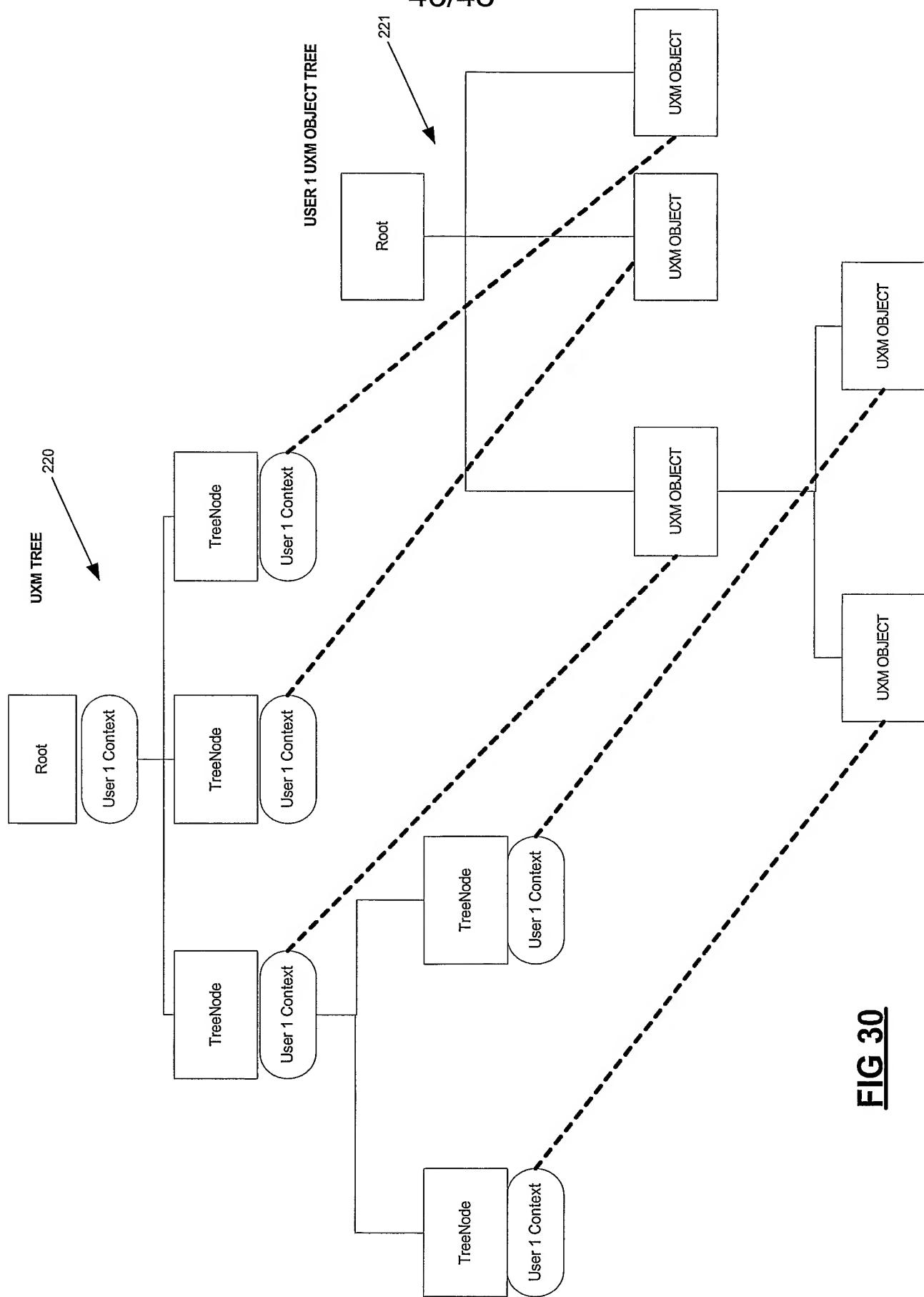
```

[CONFIG]
  [SERVICE]
    [BASE_CL_SERVICE]
      [CA_CL_SERVICE]
        [LAB1_srcApps]
<g2config>
  {GRF} firstbyte =
  [FirstByteCRM].g2config

[EXT_APPS]
  [FirstByte]
<g2config>
  {STR} authorization = false
  {STR} protocol = http
  
```

FIG 29

46/48

**FIG 30**

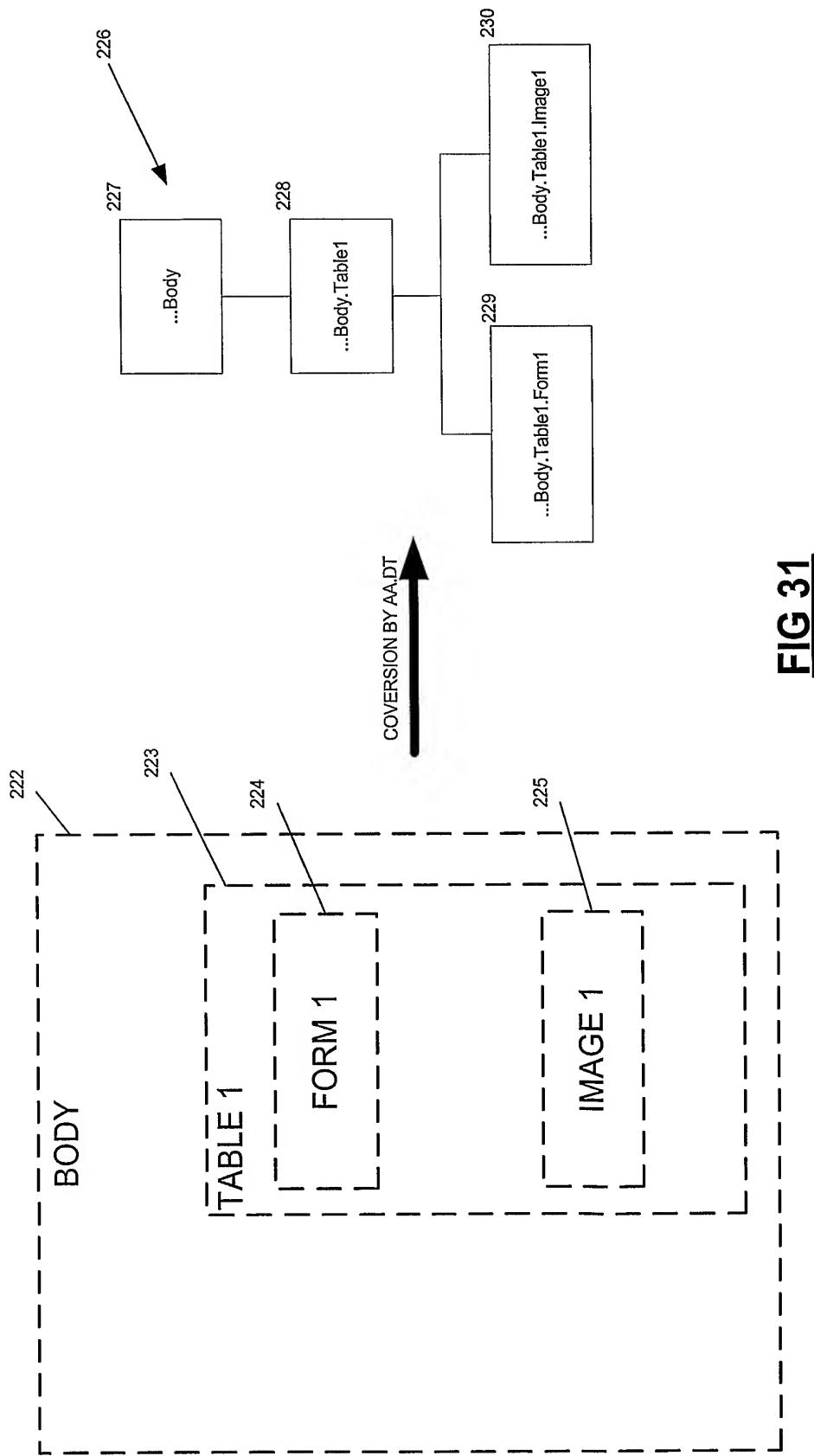
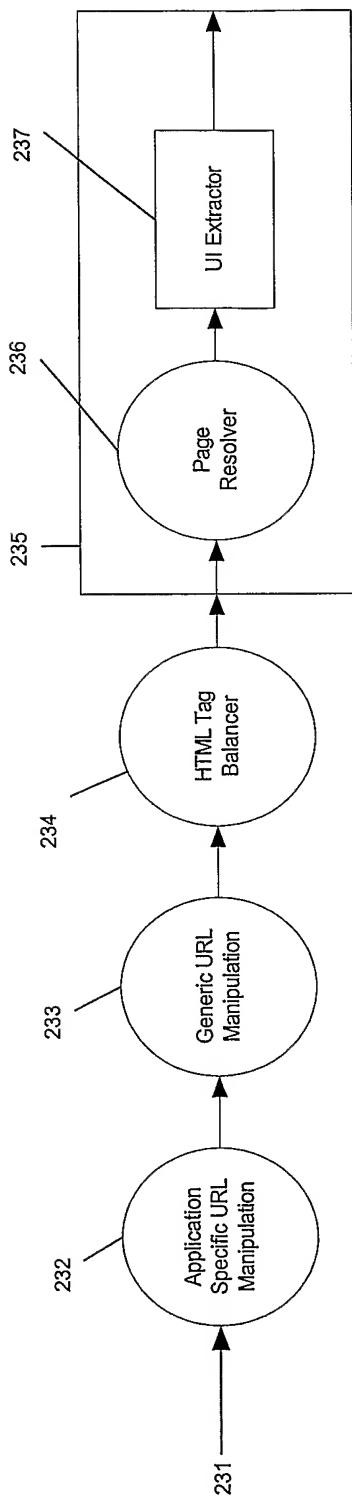


FIG 31

**FIG 32**

```

<html>
<form action="/G2/G2" method="POST">
<input type="hidden" name="APPLICATION_CLASS" value="Lab1App">
<input type="hidden" name="UXM_MODIFIED" value="true">
<input type="hidden" name="UXM_TREEID" value="Lab1App">
<input type="hidden" name="UXM_NODEID" value="CustPg">
...
</html>
    
```

FIG 33